Politecnico di Milano

FACOLTÀ DI INGEGNERIA CIVILE, AMBIENTALE E TERRITORIALE

Corso di Laurea Specialistica in Ingegneria per l'Ambiente e il Territorio

The Neighbor Search approach applied to reservoir optimal operation: the Hoa Binh case study

Candidato:
Guido Petrucci
Matricola 674821

Relatore:
Prof. Ing. Rodolfo Soncini Sessa
Supervisor:
Prof. Dragan Savic

Anno Accademico 2005–2006

The Neighbor Search approach applied to reservoir optimal operation: the Hoa Binh case study

Guido Petrucci

20th November 2006

Contents

List of Figures

Sintesi

L'uso efficiente delle risorse naturali è una delle priorità fondamentali nell'ottica dello sviluppo sostenibile. L'acqua, in particolare, gioca un ruolo chiave nel mantenimento di molti equilibri naturali ed è necessaria a numerose attività umane. Ovunque la domanda di acqua è in aumento e il suo sfruttamento diviene più intensivo, senza che ci sia un corrispondente incremento nella disponibilità. Al contrario, i fenomeni di inquinamento incidono negativamente sulle quantità attualmente sfruttabili. E' noto, inoltre, che la distribuzione sia spaziale sia temporale di questa risorsa è tipicamente disuniforme e stocastica, e ciò aggiunge ai rischi legati alla disponibilità quelli dovuti agli eventi estremi come le piene e le esondazioni. In conclusione, la gestione dell'acqua è un processo importante e complesso, che diviene ogni giorno più conflittuale.

Questa tesi tratta della gestione ottima di un serbatoio idrico. Lo scopo è quello di regolare il naturale flusso dell'acqua per ridurre il rischio di esondazione e, nello stesso tempo, per garantire la massima disponibilità della risorsa per i suoi diversi usi. L'obiettivo più generale è di inserire un nuovo tassello, in questo caso un algoritmo di ottimizzazione, nel vasto quadro dell'uso efficiente dell'acqua e delle risorse naturali.

Oggi, in molti casi, la gestione dei serbatoi è ancora basata su regole euristiche e su giudizi soggettivi del Regolatore. Conseguenza di questo fatto è che molte grandi riserve idriche nel mondo non riescono a fornire i benefici per le quali erano state pianificate (Labadie, 2004). Sfruttare al meglio le potenzialità delle strutture esistenti non rappresenta solo un obiettivo economico: rende infatti possibile stimare al meglio la necessità di nuovi serbatoi, tipicamente opere con elevati impatti sociali e ambientali (WCD, 2000).

Da un punto di vista tecnico, l'ottimizzazione della gestione dei serbatoi idrici è un problema di non semplice soluzione; i sistemi coinvolti sono infatti, normalmente, di alta dimensionalità, dinamici, non-lineari e stocastici. Inoltre, soprattutto per i laghi regolati, ma anche per gran parte dei serbatoi artificiali, gli scopi che si devono tenere in considerazione sono numerosi: oltre alla laminazione delle piene e alla produzione idroelettrica (gli obiettivi del sistema qui considerato) spesso si deve soddisfare la domanda d'acqua per l'irrigazione e per i consumi urbani e industriali, si deve garantire la navigazione ed evenutali usi turistici, si devono includere nell'ottimizzazione anche

obiettivi ecologici, come il mantenimento di deflussi minimi negli effluenti o di determinati livelli stagionali.

E' importante notare che nei problemi a molti obiettivi (MO) non esiste normalmente una soluzione ottima, ma è possibile solo determinare un insieme di alternative efficienti, la *Frontiera di Pareto* del problema. Infatti, se è possibile scartare alcune soluzioni perchè sono peggiori di altre rispetto a ogni obiettivo (*alternative dominate*), si possono trovare alternative che non sono ordinabili, in quanto non dominate da altre. Tali alternative sono dette anche *Pareto-efficienti* o semplicemente *efficienti*.

Ciò implica che la scelta della soluzione da applicare non è più direttamente il risultato dell'ottimizzazione, ma emerge da una successiva fase di negoziazione fra i portatori d'interesse (*stakeholders*), e da una fase decisionale. L'obiettivo della fase di ottimizzazione diventa quindi il supporto alla negoziazione e alla decisione, e la scelta di un algoritmo fra quelli esistenti deve essere orientata in questa prospettiva.

I metodi tradizionalmente più applicati in questo settore sono la programmazione lineare (LP) e la programmazione dinamica (DP). Il primo è particolarmente efficace per sistemi di grandi dimensioni, essendo molto veloce ed efficiente; d'altra parte richiede che il sistema da ottimizzare sia lineare (o linearizzabile), anche per quel che riguarda vincoli e obiettivi.

La programmazione dinamica non impone, invece, così forti ipotesi sul sistema (che deve essere, però, discretizzato) ed è particolarmente adatta a problemi con decisioni ricorsive come, appunto, la gestione ottima dei serbatoi. Il limite principale di questa tecnica risiede nella dimensionalità: il tempo e le risorse necessarie per il calcolo crescono esponenzialmente con il numero di stati in cui il sistema può trovarsi[1].

Entrambi i metodi descritti operano su problemi a singolo obiettivo, ed è quindi necessario l'uso di tecniche per ridurre l'originario problema MO in una serie di problemi di questo tipo. Diverse sono le opzioni disponibili, brevemente presentate nel paragrafo 1.2. Una delle più diffuse, il *metodo dei pesi*, consiste nell'aggregare gli N diversi obiettivi con una media pesata; variando il vettore dei pesi impiegati (w) si possono ottenere diverse soluzioni efficienti. Si può dimostrare che, se la Frontiera di Pareto è convessa, l'esplorazione dello spazio dei pesi W corrisponde all'esplorazione della Frontiera di Pareto stessa[2].

Il principale problema di questo approccio è che il modo in cui i vettori vengono scelti in W non viene specificato. Nonostante esistano criteri di campionamento, questo fatto diventa sempre più limitante all'aumentare del numero di punti della Frontiera di Pareto e del numero N di obiettivi. Infatti, dati due vettori $w^1, w^2 \in W$, (i) non è possibile sapere a priori se essi individuano due punti

[1] Una trattazione più ampia di queste e delle altre principali tecniche di ottimizzazione, insieme ad alcuni esempi di applicazione alla gestione dei serbatoi idrici, viene presentata nei paragrafi 1.1 e 1.1.1.

[2] Si veda, a questo proposito il paragrafo 2.2, e in particolare il corollario 1.

distinti nello spazio degli obiettivi e, (ii) se individuano due punti distinti, non è possibile sapere se e quanti punti sono compresi fra i due.

L'algoritmo qui proposto, il *Neighbour Search*, è una possibile soluzione a questo problema. Infatti, per un'ampia classe di problemi[3], esso sfrutta alcune proprietà geometriche della Frontiera di Pareto per determinarla *completamente*. L'approccio consiste nel partire da un punto noto (determinato, ad esempio, tramite la DP) e trovarne i "vicini" (*neighbours*) nello spazio dei pesi e sulla Frontiera di Pareto; dopodichè, iterando il procedimento, si procede nell'esplorazione.

Due sono i principali concetti su cui si basa l'algoritmo:

La *Q-function* di una politica φ, per una coppia stato-controllo (x, a), esprime i costi-futuri attesi ottenibili partendo dallo stato x, applicando il controllo a al primo passo temporale e la politica φ per il resto dell'orizzonte temporale. E' quindi una funzione $Q(\varphi, x, a) : X \times A \to \mathbb{R}^N$, una volta definita la politica φ. Applicare il metodo dei pesi, dato un vettore $w \in W$, equivale a trovare la politica φ^* tale che[4]:

$$\forall x, \ \min_a \langle Q(\varphi^*, x, a), w \rangle = \langle Q(\varphi^*, x, \varphi^*(x)), w \rangle \tag{1}$$

Il *Preference set.* In un sistema discreto, ogni politica ottima φ^* (a cui corrisponde un punto s^* della Frontiera di Pareto) è tale non per un singolo vettore dei pesi w, ma per un insieme di pesi. Più precisamente definiamo il *preference set* $W(s^*)$ di un punto s^* l'insieme dei pesi per i quali tale punto è l'ottimo. Come si può intuire, questo è equivalente a dire che se e solo se $w \in W(s^*)$, la minimizzazione descritta dall'equazione 1 ha come risultato la politica corrispondente a s^*.

Proprio basandosi su quest'ultima proprietà l'algoritmo può, data una politica ottima φ^* e la sua Q-function, determinarne il preference set e i "neighbours" $\varphi_1^*, \varphi_2^*, \ldots$. A questo punto è sufficiente calcolare le Q-function di queste nuove politiche e iterare il procedimento finchè non si è coperto l'intero insieme W.

Il risultato di questo processo è, come già accennato, l'intera Frontiera di Pareto del problema. Si può immaginare che nel caso di sistemi molto grandi, in particolare con una Frontiera costituita da un elevato numero di punti, l'incremento dei tempi di calcolo faccia optare per una esplorazione solo parziale della Frontiera stessa. Il NS si presta anche a questo tipo di analisi, visto che può essere vincolato ad operare solo in sottoinsiemi di W.

La forma più ampia e generale del Neighbour Search è descritta nell'articolo (Dorini et al., 2006a), dove ne viene dimostrata l'efficacia. Viene anche presentato un esempio di come esso possa essere

[3] L'algoritmo come viene enunciato qui di seguito e, più esaustivamente, nel capitolo 2, lavora su *Markov Decision Processes* (MDP) finiti, e il sistema deve essere quindi discretizzato, finito e stocastico. Sostanzialmente queste ipotesi sono le stesse imposte dalla Programmazione Dinamica.

[4] L'operatore $\langle x, y \rangle$ è il prodotto scalare; si vedano le definizioni date nel paragrafo 2.1.

impiegato per ottimizzare un generico MDP. Il lavoro descritto in questa tesi, invece, rappresenta la prima applicazione di tale approccio a un caso reale.

Il problema che è stato preso in esame è l'ottimizzazione della gestione del serbatoio di Hoa Binh. Si tratta di un grande lago artificiale nel nord del Vietnam, che capta le acque del fiume *Da*, principale affluente del fiume Rosso (figura 3.1). La diga è operativa dal 1990, con i due obiettivi principali di laminazione delle piene a valle e di produzione idroelettrica. Tali obiettivi sono conflittuali durante la stagione delle pioggie (da giugno a settembre) quando, a causa del regime dei monsoni, è concentrato l'80% delle precipitazioni annuali[5].

Infatti, se si invasano grandi quantità d'acqua si aumenta la disponibilità di energia elettrica (l'impianto idroelettrico di Hoa Binh soddisfa il 40% del fabbisogno dell'intero Paese) ma, nello stesso tempo, si riduce la capacità di laminazione se si presenta un evento di piena. Si consideri che la costruzione del serbatoio è stata motivata proprio dalla vulnerabilità alle piene della ricca e popolosa regione del delta del fiume Rosso, che comprende la capitale del Vietnam Hanoi.

L'attuale regolazione usa una legge di controllo che dipende dal periodo dell'anno, dal livello delle acque nel serbatoio e da una previsione del livello ad Hanoi per il giorno successivo. Per realizzare tale previsione vengono impiegate le misure di portata realizzate sugli altri due principali affluenti, il *Thao* e il *Lo*, oltre che il rilascio da Hoa Binh.

Il serbatoio e la sua legge di controllo sono già oggetto di due studi svolti da Le Ngo et al. (2006) e Madsen et al. (2006). Gli autori hanno realizzato un modello idraulico del sistema usando il software MIKE-11 (DHI, 2005), e hanno ottimizzato alcuni parametri dell'attuale regolazione tramite un algoritmo genetico, attraverso uno schema di simulazione-ottimizzazione su scenari deterministici. I dati riguardanti il sistema impiegati in questo lavoro, così come i risultati dei due studi, sono stati gentilmente forniti dagli autori citati.

Il modello del sistema che è stato realizzato (descritto in dettaglio nella sezione 3.2) comprende tre variabili di stato:

1. l'invaso del serbatoio s, la cui dinamica è regolata da un'equazione di bilancio di massa integrata su passi di 20 minuti (equazione 3.4),

2. il livello delle acque ad Hanoi h, che viene determinato ad ogni passo da una rete neurale i cui ingressi sono i rilasci dal serbatoio, le portate dei due affluenti non regolati e il livello al passo precedente (si veda il paragrafo 3.2.5),

3. la configurazione degli scarichi della diga b, necessaria per poter tenere conto di alcuni vincoli imposti ai rilasci ammissibili (si veda il paragrafo 3.2.4).

[5] Una descrizione dettagliata del sistema è presentata nel paragrafo 3.1.

Sono inoltre presenti tre disturbi stocastici, di distribuzione lognormale, che rappresentano le portate negli affluenti. Sono indicati come w^{Da}, w^{Thao} e w^{Lo}.

Il controllo del sistema è svolto tramite la variabile discreta a, che rappresenta il *"numero di rilasci aperti alla fine del passo temporale"*. Tale formulazione, che può apparire inusuale, ben si presta a rappresentare le decisioni di rilascio da Hoa Binh: tre diversi canali sono infatti disponibili per i deflussi, cioè le otto turbine, i dodici scarichi di fondo e i sei sfioratori. E' quindi più agevole definire quali di questi rilasci aprire o chiudere, piuttosto che una portata o un volume d'acqua. Il passo temporale scelto fra una decisione e la successiva è di 48 ore (paragrafo 3.2.2).

Per definire le curve di rilascio dei diversi canali sono state usate interpolazioni dei dati disponibili, così come è stato fatto per le curve di produzione delle turbine.

L'attuale politica di regolazione è rigida per quanto riguarda il massimo invaso ammissibile: quando si raggiungono i 5 m dal livello critico per la diga, è obbligatorio rilasciare una portata almeno pari all'afflusso. Per evitare di limitare il modello in questo modo, pur senza creare rischi per il serbatoio, si è trasformato il vincolo in un obiettivo che penalizza gli invasi troppo elevati. Questo, insieme agli altri due costi considerati (esondazioni di valle e deficit idroelettrico), è esplicitato nel paragrafo 3.2.6.

Una volta definito il modello, se ne è ricavato il Markov Decision Process corrispondente (questa operazione è descritta nella sezione 3.3). Questo comporta la discretizzazione dello spazio di stato X, dei controlli A e degli afflussi W e la simulazione del sistema per ogni possibile elemento di $X \times A \times W$. Il risultato è la probabilità di transizione di stato $p(y|x, a)$ e il vettore di costi associato $g(x, a)$.

Grazie ai risultati delle ricerche di Madsen e Le Ngo, è stato possibile procedere a una validazione della struttura della legge di controllo implementata nel modello. Infatti, se si confronta quest'ultima con quella attualmente applicata, appaiono alcuni differenze sostanziali. In particolare:

1. il passo temporale delle decisioni è di 48 ore invece che di 6 (intervallo fra le misure),
2. le decisioni dipendono solo dai livelli nel serbatoio e ad Hanoi, e non anche dal periodo dell'anno e dagli afflussi,
3. si usano 27 controlli ammissibili, selezionando solo una parte di quelli realmente disponibili,
4. la politica è definita solo per le classi di stato, e quindi su un insieme discreto,
5. d'altra parte la "forma" della politica è libera, e non vincolata da una funzione di cui si ottimizzano solo pochi parametri come nel caso di Madsen e Le Ngo.

Ci si è quindi posti il problema di verificare se una legge di controllo definita su queste basi può ottenere performance almeno analoghe a quelle della regolazione attuale. Tale validazione è stata effettuata (si veda la sezione 3.4) e i risultati sono mostrati in figura 3.12. Come si può vedere ciò conferma le scelte modellistiche fatte.

Una volta ricavato il MDP e verificata la struttura della legge di controllo, si è potuto applicare il Neighbour Search al sistema. Questa ottimizzazione ha dato come risultato una frontiera di Pareto costituita da più di un milione di punti, di cui è rappresentato un dettaglio in figura 4.1 e le tre proiezioni bidimensionali, rispetto alle diverse coppie di obiettivi, nelle figure 4.2, 4.3 e 4.4.

Per fornire degli esempi delle possibilità offerte al processo decisionale dalla conoscenza di tutte le alternative ottime, nel capitolo 4 è illustrata una serie di politiche ricavate dalla Frontiera di Pareto usando diversi criteri:

1. punti estratti con una ricerca su tutta la Frontiera basata sui valori degli obiettivi (in questo caso, i minimi per ciascuno di essi),
2. punti estratti sulla base della posizione nella Frontiera, selezionati visivamente (in particolare si sono estratti due punti agli estremi della zona di basso tasso marginale di sostituzione fra gli obiettivi V^{HyPow} e V^{UFlood}),
3. punti estratti tramite movimenti vincolati nello spazio degli obiettivi. Più precisamente, partendo da un punto dato, si sono ottenuti dei punti spostandosi su delle curve "iso-obiettivo" (si veda la figura 4.10);
4. punti estratti perchè "vicini" di un punto dato.

Alcune delle politiche ricavate sono mostrate nell'appendice A, sia esplicitamente sotto forma di tabelle, sia simulate sullo scenario storico di afflussi, per valutarne i comportamenti.

Quello che è importante sottolineare di questi esempi sono le diverse possibilità che vengono offerte, ai portatori d'interessi e al decisore, per esplorare la Frontiera di Pareto. In particolare, questo tipo di analisi può essere *istantaneo* e quindi condotto in tempo reale durante la negoziazione, dato che le politiche sono note a priori[6].

Nel caso il sistema sia troppo grande per calcolarne tutta la Frontiera indistintamente, come si è detto il NS permette di limitarsi a sottoinsiemi di W.

Ciononostante, non necessariamente i portatori d'interesse o il decisore sono in grado di indicare una zona significativa dello spazio dei pesi (si pensi, per esempio, ad una simile operazione in uno spazio a 6 dimensioni). Di conseguenza, una delle più promettenti aree da investigare nelle future ricerche sul NS è la possibilità di rendere l'algoritmo *direzionabile*, cioè permettere un'interazione tale da poter dirigere l'algoritmo verso le parti più promettenti della Frontiera di Pareto. Questo

[6] E' chiaro che, se vengono richieste le simulazioni delle politiche o altre elaborazioni, bisogna tenere in considerazione il tempo per produrle. Tuttavia questo è di norma ampiamente inferiore al tempo necessario per individuare e calcolare le politiche ricercate.

dovrebbe essere possibile sia fissando una direzione di riferimento costante, sia permettendo un processo iterativo di confronto col decisore.

Queste ultime osservazioni suggeriscono che, per approfittare di tutte le potenzialità e della flessibilità del Neighbour Search, potrebbe essere necessario sviluppare un sistema di supporto alle decisioni (DSS) dedicato.

Infatti, la differenza chiave fra gli approcci tradizionali e il NS è che con quest'ultimo si trovano, insieme ad ogni politica, delle informazioni aggiuntive sulla topologia della Frontiera di Pareto. Tali informazioni possono essere, come si è in parte mostrato, di notevole utilità per la negoziazione e per il processo decisionale. Appare chiaro che una riflessione più approfondita sui modi in cui queste informazioni possono essere sfruttate a fondo può portare a ulteriori sviluppi e benefici.

In conclusione, da questa prima applicazione a un caso reale, emerge che l'approccio del Neighbour Search può effettivamente dare un contributo per raggiungere un più efficiente, consapevole e partecipato uso dell'acqua e, in generale, delle risorse naturali.

Introduction

In the actual world trend to a more sustainable development, one important role is played, at any scale, by the efficient use of natural resources. There are three main reasons: natural resources are valuable because they are scarce, and although renewable, they not always have a sufficient replacement rate. They are socially important, as access to such resources is fundamental for human life. Last but not least they have a great importance for the environment, as they maintain ecological equilibriums at both local and global scale. Therefore, using natural resources wisely must be one of our main priority.

In addition to all the features stated above, freshwater has some distinctive traits: it is necessary to all life-beings, and it is involved in most of the human activities. It is not always and everywhere scarce but it is unequally distributed, spatially and temporally. This natural stochasticity creates risks linked both to availability of supplies, and to natural disasters such as flooding. The management of freshwater resources draws from a broad spectrum of disciplines, ranging from production and treatment, to storage and distribution. For all those reasons, water management deserves great attention and involvement, including a strong and continue research effort to improve our use of the water resources.

This thesis focuses on reservoir optimal operation. It deals with the issues of regulating water natural flow in order to reduce risks of flooding while maximizing availabilty of water for various purposes.

At present, reservoir operation strategies are mainly based on heuristic procedures and/or on subjective judgments of the operator. One consequence of this fact is that many large storage projects worldwide are not providing the levels of benefits they were planned for (WCD, 2000). Finding a way to better exploit the potential of such existing reservoirs is not only an economic target, but enables to estimate the real need for further reservoirs, which are structures with high environmental impact.

Reservoir operation optimization is a challenging problem. The systems involved are normally high-dimensional, dynamic, non-linear and stochastic. Further, most of the reservoirs are multi-purpose: they perform flood control and also provide water for hydropower production, irrigation, urban and industrial consumption, navigation and others. Since many actors and stakeholders are typically involved, with different demands and purposes, optimization methods capable of handling multiple objectives are usually required. Several solution techniques have been developed and applied to reservoir optimization, such as Linear Programming (LP), Dynamic Programming (DP) or heuristic methods like Evolutionary Algorithms (EA). These alternatives will be presented in chapter 1.

An important point to underline is that, in MO problems, an optimal alternative does not exist, but only a set of efficient alternatives can be found. It means that the final decision is not the result of the optimization process, but it emerges from a negotiation phase that involves all the stakeholders. So, the purpose of the optimization is no more to find the better solution to the problem but to support, in the better possible way, the negotiation and the decision-making. The evaluation and the choice of an optimization method among the existing ones must be carried out with this perspective.

In this work, the Neighbor Search (NS) optimisation proposed by Dorini et al. (2006a) is applied for the first time to the problem of multi objective optimization of reservoir operation. We believe that this algorithm has wide application possibility in decision-support systems (DSS), that will benefit from it. In a wide range of cases[7], it allows the exhaustive exploration of the set of efficient alternatives (the *Pareto-set*), providing the negotiation stage with a complete knowledge of optimal solutions. This exploring capability can also be used in an iterative search process, where stakeholders and decision-makers can indicate the exploring directions.

The case-study to which the NS algorithm is applied is the Hoa Binh reservoir optimization. Hoa Binh is a large reservoir in northern Vietnam, in operation on the Red river since 1990 with the two main purposes of flood control and hydropower generation. These purposes are conflicting during the rain season (from June to September) when because of the monsoon climate, is concentrated 80% of the annual rainfall.

Storing large amounts of water will guarantee power supply availability during the dry season (Hoa Binh hydropower plant produces 40% of the whole Vietnam electric supply). On the other hand, if a major flood event occurs, unused storage capacity can be exploited to reduce flooding damage in the Red river delta region. The Hoa Binh dam was built to protect this rich and densily-populated area, including Vietnam capital Hanoi.

The reservoir is currently operated using a rule-curve depending on the period of the year, on the water-level in the reservoir and on a forecast of the water-level in Hanoi.

Hoa Binh reservoir and operation rule have been already studied by Le Ngo et al. (2006) and

[7] It can be applied to all systems for which a Markov Decision Process approach is suitable.

Madsen et al. (2006). They produced an hydraulic model of the system using the MIKE-11 software (Le Ngo et al., 2006; DHI, 2005), and they optimized the operation rule-curve using a genetic algorithm in a simulation-optimization framework (Madsen et al., 2006). The data pertaining the model of the system used in this thesis, as well as the results used for comparison purposes, was kindly provided by authors of these studies.

In chapter 1 the existing techniques used in multi-objective reservoir optimization are shortly described. In chapter 2 is discussed the Neighbor Search algorithm. In chapter 3 the Hoa Binh system and the corresponding models are presented, while in chapter 4 the system optimization is detailed. Chapter 5 concludes this thesis with some observation and discussion on future work.

Chapter 1

Multi-objective optimization of reservoir operation

1.1 Review of methods for reservoir operation optimization

Several techniques have been applied to reservoirs operation optimization, attempting to improve their performances. Few basic methods are generally used, although many variants have been implemented in order to overcome some limitation of the method or to deal with some particular application. A state-of-the-art review, particularly focused on multi-reservoirs systems, is presented in (Labadie, 2004). A wide sample of real-case applications of optimization techniques to both single- and multi-purpose reservoir operation can be founded in (Wurbs et al., 1985).

Historically, one of the most favored optimization technique is *linear programming* (LP). It has some notable advantage: several highly-efficient solving algorithms are available; it is able to solve extremely large-scale problems; it converges to globally-optimal solutions; theory for sensitivity analysis is well-developed.
Linear programming is particularly useful for large multi-reservoirs systems. Hiew et al. (1989) applied deterministic LP for the optimization of the Colorado-Big Thompson eight-reservoir system, obtaining optimal storage guide-curves. In this particular case, a close-to-linear behavior of the system allowed to obtain good results. Actually the main limitation of this method is that the model, including constraints and objectives, must be linear or linearizable.

Several adaptations of the method have been implemented to bypass this strong hypothesis. For example, in *separable programming*, piecewise linear approximations of nonlinear functions are used. On the other hand, such extensions make the problem dimensions grow and the solver efficiency decrease. Moreover, for some case the convergence to a global optimum can not be guaranteed. These methods have been applied to some real case, such as the multi-reservoir Metropolitan Adelaide water supply system in Australia, by Crawley and Dandy (1993).

Despite this adaptations, many reservoir optimization problems cannot be realistically modeled as linear or piecewise-linear function. Typically hydropower generation functions can hardly be approximated as linear, as the head effects on production are strongly nonlinear. A possible solution to this issue can be founded through *non-linear programming*, that requires only differentiability of the model's equations.

Also for this method several implementations exist. The far most efficient, according to Hiew (1987) and Grygier and Stedinger (1985), is the so-called *successive linear programming* (SLP), based on a iterative linearization-LP loop. The main disadvantage of SLP, as pointed out by Bazarra et al. (1993) is that this method is not guaranteed to converge.

Barros et al. (2003) applied the SLP technique to the Brazilian hydropower system, one of the largest in the world. This study confirmed the good performance of the method, in terms of accuracy and computational efficiency. Other NLP variants have been applied to the four-reservoir Zambezi river system by Arnold et al. (1994) and to the Highland Lakes of the Lower Colorado River basin by Unver and Mays (1990).

Besides LP and NLP, which are defined for deterministic problems, other methods have been adopted for solving stochastic problems. Kall and Wallace (1995) propose a two-stage optimization, where the objective to minimize is the cost from first-stage decision plus the expected future costs, evaluated over several scenarios, each with an assumed probability of occurrence. Following implementation of the first-stage decisions, the problem is reformulated starting with the next time-step decisions and solved over the remainder of the operational horizon. The difficulty with this formulation is that, if many possible scenarios are taken into account, the resulting problem can become too resources-expensive. Improved versions of this technique coupled with LP are used by Jacobs et al. (1995) to optimize the Pacific Gas and Electricity hydropower system in northern California, and by Seifi and Hipel (2001) for the Great Lakes Reservoir system.

The last 10 years have witnessed significant advances in the development of heuristic-based optimisation methods, and in particular Evolutionary Algorithms (EAs), which work by repeatedly sampling the search space, guided by the information collected during the search process and held as a memory in the form of a population of solutions. EAs are derivative-free global search methods, and they were shown to work well on nonlinear, nonconvex, and multimodal problems (Back et al., 1997). One of the earliest applications of EA in reservoir control was the work by Esat and Hall (1994). They applied a GA to a four reservoir problem in order to maximize the benefits from power generation and irrigation water supply subjected to constraints on reservoir storages and releases.

Sharif and Wardlaw (2000) used Genetic Algorithms (GA) to optimize a real multireservoir problem - Brantas Basin in Indonesia. They considered 4 case studies: (i) maximizing hydropower returns; (ii) maximizing hydropower and irrigation returns; (iii) same as (ii) but including a future water resources development scenario; and (iv) same as (iii) but including more reservoirs in the system.

Generally, for EAs dealing directly with multi-objective problems (often addressed as MOEAs), convergence to Pareto-optimal solutions can not be guaranteed. For some particular techniques, however, proofs of convergence are showed by Rudolph (1998) and Hanne (2001, 1999). Laumanns (2003) demonstrates that although the convergence (to the limit) can be assured by such techniques, it is not guaranteed a good distribution of the solutions.

1.1.1 Markov Decision Processes and Dynamic Programming

All the methods introduced share a limitation about time-steps: increasing the number of time-steps (i.e.: optimizing over a longer time-horizon, or considering shorter intervals between decisions) bring to an exponential augmentation of calculus-time. For this reason, to keep low the time-steps number, most of the applications cited optimize monthly decisions. If such a period can be adapted for some cases, in many other it will be too long, giving an information insufficient to actually manage efficiently the reservoir system. For example, in a flood event, a regulation should be taken hourly, in order to effectively minimize damages. In such a situation the operator will not be helped by the knowledge of the monthly average flow he or she has to release.

A technique which can overcome this problem, for this reason one of the most popular for reservoir operation optimization, is Dynamic Programming (DP, for an extensive discussion, see Bertsekas (1995)). This method exploits the sequential nature of reservoir operation and reduces the optimization-time dependence by the number of time-steps from exponential to linear. The key strategy of the DP is to split the whole optimization problem in a series of one-stage (i.e.: time-step) interrelated sub-problems, and then to solve them sequentially. A second important advantage of DP is that objective functions must be defined under fairly weak conditions.

Moreover DP optimization, through expected-cost functions, can take in account the stochastic nature, typical of reservoir control problems[1]. This stochastic case can be usefully expressed as an optimization program for a Markov Decision Process. The general definition of MDPs will be given in chapter 2, while a detailed discussion of their use for water reservoir optimization can be found in Lamond and Boukhtouta (2002).

Labadie (1993) applied DP to the Valdesia Reservoir in the Dominican Republic. Terry et al. (1986) compared optimal DP solutions with traditional rule curves for the Brazilian system, obtaining substantially improved results. Several other researchers applied stochastic DP to reservoir operation, such as Stedinger et al. (1984) that worked on the High Aswan dam system, Huang et al.

[1] This is why in this work no distinctions are made, as some authors do, between DP and stochastic DP (SDP). In what follows it will be used "DP" to refer to this latter case.

(1991), Vasiliadis and Karamouz (1994). In (Tejada-Guibert et al., 1993, 1995) different DP approaches are applied to the Shasta-Trinity system, a multi-reservoir subsystem of the Central Valley Project, California.

The main issue of using DP is the so-called *curse of dimensionality*: optimization time depends exponentially on the number of state-class. It involves that for a multi-state system (typically, but not always, a multi-reservoir system), or for a finely-discretized system, the necessary computation effort can overcome disposable resources. Many variations of the basic algorithm have been implemented to overcome this problem, but a completely satisfying and general solution is not yet founded.

Examples of these approaches are the *Differential Dynamic Programming* (DDP), *extended linear quadratic Gaussian* control (ELQG) and *Neural Dynamic Programming* (NDP).

DDP, developed by Jacobson and Mayne (1970), searches analytical solutions of the problem, without discretization of the states. This requires to impose additional strong conditions on the model's equation, and resemble to a NLP formulation, but with the strong advantage of stage separation proper of DP. DDP has been applied to the Mad River system in California by Jones et al. (1986).

ELQG (Bertsekas, 1995) is based on the same idea of a continuous state and an analytical solution. In this approach the state-variables are replaced by their mean and variance, and assumptions on their probability distributions are required. This method has been implemented for the High Aswan Dam by Georgakakos (1989), obtaining more efficient reservoir operation policies than Stedinger et al. (1984).

NDP is a different approach that approximate the *cost-to-go* function[2] with an Artificial Neural Network. The conditions to apply this method are weak and its convergence to a good solution is guaranteed (see Bertsekas and Tsitsiklis (1996)). Application of this method to reservoir operation is discussed in (De Rigo et al., 2001) and tested on the three-reservoir Piave Project, Italy (Soncini Sessa, 2004).

1.2 Multi-objective optimization

With the exception of MOEAs, optimization techniques described above deal with a single objective. This require to use methods to extend the application of such methods to MO problems.

In a MO contest, one optimal solution can not, generally, be founded. The reason is that a direct comparison between two alternatives will not always issue in an ordering relation, as an alternative can perform better than the other for an objective and worse for another. In the case a order can

[2] Obtaining this function, in DP, is dual to finding the optimal policy.

be founded through the comparison (i.e.: solution A is better than solution B regarding to all the objectives) it will be said that *A dominates B* and that *B is dominated*. In general, the solution of a MO problem is the set of all the non-dominated alternatives, also called *Pareto-set*. The alternatives forming the Pareto-set are also called *efficients* because, for each of those, no solution can be found that improves an objective without degrading another.

A formal definition of domination and Pareto-set is given in chapter 2. Here is just underlined that several methods are available, to reduce a MO problem in order to solve it with one of the single-objective techniques just described.

The *constraint method* requires to transform $N-1$ of N objectives in constraints, and to define a threshold for each of them. Then the SO problem is solved. Varying the set of thresholds, different points of the Pareto-set can be calculated. Yeh and Becker (1982) applied this system to study the trade-off between hydropower generation and water supply for the Central Valley Project in California.

A second way is the *weights method* that consists in aggregate all the objectives in a single scalar through a weighted sum. Varying the coefficients of the sum allows to find different points of the Pareto-set. A comparison between this method and the preceding one was performed by Ko et al. (1992) on a 4-objectives study of the Han river reservoir system, Korea. The conclusions of this study was that the weights method was preferable for large numbers of objectives. The negative aspect of this latter method is that it works less effectively on concave Pareto-fronts. In such cases, the method can not find all the optimal policies.

Another technique is the so-called *goal-programming*. It has been applied to the TVA's reservoir system by Eschenbach et al. (2001). This method requires a hierarchic ordering of the objectives. Each objective function is then minimized individually following the ordering. All the alternatives that satisfy a goal posed for the objective are conserved, and passed to the further optimization (i.e. for the next objective of the hierarchy). This method works well only if many alternatives pass through each level and, further, some knowledge about the preference structure of the decision-maker is necessary to obtain a good result.

Chapter 2

The Neighbour Search for constructing Pareto sets

The Neighbour Search approach (NS) is a methodology for exploring Pareto sets in multi-objective frameworks, in where performance sets are convex polytope. Typical problems that can be effectively addressed by NS are multi-objective linear programming and multi-objective Markov Decision Processes. In this chapter NS and the application for MDP is described. The reader that is interested in more details about the theoretical bases, and the proofs of the propositions, is referred to the article (Dorini et al., 2006a) and the thesis (Dorini, 2006).

2.1 Notation and basic definition

In what follows, there are few words about the notation adopted throughout the chapter. Vectors of $\mathbb{R}^N$ are columns, upper indices $x^1, x^2, \ldots$ correspond to different vectors, whilst lower indices are the vector components: $x = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^N$. The scalar product $\sum_{k=1}^{N} x_k y_k$ is denoted with $\langle x, y \rangle$; 'Def.' is the abbreviation for 'Definition'.

Def. 1. A set $D \in \mathbb{R}^N$ is a convex set, iff for every pair of points $x, y \in D$, the whole segment $\theta x + (1-\theta)\, y, \theta \in [0,1]$ belongs to D.

Def. 2. A convex polyhedrical set D, or simply *polyhedron*, is the intersection of $I < \infty$ halfspaces in $\mathbb{R}^N$, namely:

$$D = \bigcap_{i=1}^{I} \left\{ x \in \mathbb{R}^N | \left\langle x, v^i \right\rangle \leq b^i, v^i \in \mathbb{R}^N, b^i \in \mathbb{R} \right\}$$

Bounded polyhedra are called *(convex) polytopes*.

The *dimension* of a polytope $D \in \mathbb{R}^N$ is the dimension of the space H given by the intersection of all hyperplanes in $\mathbb{R}^N$ that contain D. Obviously, if D is not a singleton, H is always a hyperplane of various dimensions. In literature, H is often denoted as the *affine hull* of D. A $d-$polytope, is a polytope with d dimensions.

Def. 3. A convex subset F of a convex set D is called *extreme* if representations $\theta x + (1-\theta)\,y \in F, \theta \in [0,1]$ is possible only if $x, y \in F$. A convex subset F of a convex set D is called *face* if there is a hyperplane $H \subset \mathbb{R}^N$ supporting D in F, namely: $H \cap D = F$.

Clearly, a face is a polytope itself, and vice versa. A $d-$polytope is actually a $d-$face; $(d-1)-$faces are called *facets*; $1-$faces are called *edges* and $0-$faces are *vertices*. The faces $F^2, F^3, \ldots$ of a polytope D can be (partially) ordered by the inclusion. A face $F \subset D$ is said to be a maximal face if it is not a strict subset of any face F^j.

Def. 4. The convex hull of a set $S \in \mathbb{R}^N$, denoted with $conv\,(S)$, is the intersection of all convex sets that contain S. In case of finite $S = \left\{s^1, \ldots, s^K\right\}$, the corresponding convex hull is:

$$conv\,(S) = \left\{\sum_{k=1}^{K} p^k s^k | p^k \geq 0, \sum_{k=1}^{K} p^k = 1\right\}$$

It can be proved (McMullen and Sherphard, 1971, pp. 43-47) that the convex hull of a finite set of points is a convex polytope, and that conversely, a convex polytope is the convex hull of a set of points. A very important relationship between a finite set of points S and its convex hull $D = conv\,(S)$, is that for every supporting hyperplane H, the corresponding face $F = D \cap H$ can be derived by S in the following way: $F = conv\,(S \cap H)$.

2.2 Description of the Neighbour Search

Suppose that a decision maker has K possible decisions, and that a decision k is associated to a vector of performances (losses)

$$s^k = \left(s_1^k, s_2^k, \ldots, s_N^k\right) \in \mathbb{R}^N$$

Different decisions lead to different performances; the performance set S is a set of points of $\mathbb{R}^N$. Decisions can also be randomized: let p^k be the probability for the k-th decision to be taken, the adopted performance of such randomized decision is the expectation $\sum_{k=1}^{N} p^k s^k \in D$, where the performance set D is a convex polytope of $\mathbb{R}^N$, generated by $conv\,(S)$. For the sake of comparing different solutions, points in D are partially ordered with respect of the *dominance*. A point $x \in \mathbb{R}^N$ is dominated by a point $y \in \mathbb{R}^N$, if vector $x - y$ has non negative components, namely

$$\begin{aligned} x_i - y_i &\geq 0 \qquad \text{for every } i \in \{1, \ldots, N\} \\ x_i - y_i &> 0 \qquad \text{at least for one } i \in \{1, \ldots, N\} \end{aligned}$$

A decision a is said to be dominated by a decision b if the performance $x^b \in D$ dominates performance $x^a \in D$. Obviously, there is no reason for a decision maker (DM) to prefer a to b; and when it comes to the whole performance set D, there is no reason for the DM to consider any options whose performance is dominated by some other point of D.

Def. 5. A point $x \in D$ is called Pareto optimal (or efficient or minimal), if it is not dominated by any point in D. Collection of all Pareto points of a convex set D is denoted as $Par\,(D)$.

Decisions whose performance belongs to $Par\,(D)$, are called *Pareto optimal*. Only Pareto optimal decisions are of interest of the DM; hence, the main objective of the research presented in Dorini et al. (2006a), was to develop a methodology for constructing such Pareto set. A common approach for performing such task, is to solve several scalar programs in the form $\langle x, w\rangle \to \min_{x\in D}$, where w is a vector of the space:

$$W = \left\{ w \in \mathbb{R}^N | \sum_{i=1}^{N} w_i = 1, w_i > 0 \right\}$$

Note that W is a polytope. It is well known that such scalar optimization leads to Pareto optimal solutions; furthermore, for a convex D, the set $Par\,(D)$ can be entirely explored by varying the vector w (see corollary 1). Essentially, the exploration of $Par\,(D)$ corresponds to the exploration of W; such approach is often called the *weighting method*. The problem of such approach, is that the way the vectors should be selected from W is not specified. As consequence, there are no clear directions for the exploration of $Par\,(D)$, and this becomes more and more a limit when the number of options K and the number of objectives N increases. The Neighbour Search is a possible solution of such problem; one of the main concept behind NS, is the *preference set*.

Def. 6. Given a point $x^* \in D$, a vector $w \in W$ is a *preference vector* for x^* if

$$\langle w, x^*\rangle = \min_{s\in S} \langle w, s\rangle = \min_{x\in D} \langle w, x\rangle$$

The collection of all preference vectors for x^* is called *preference set*, and it is denoted with $W\,(x^*)$.

Proposition 1. (a) If there is a point $x^* \in Par\,(D)$, then there is a vector $w \in W\,(x^*)$ such that $w \in W$. (b) If there is a point $x^* \in D$, and a vector $w \in W$ such that $w \in W\,(x^*)$, then $x^* \in Par\,(D)$

A straightforward consequence of Proposition 1, is the following corollary

Corollary 1. (a) $Par\,(D) = \bigcup_{w\in W} \{x^* | \langle w, x^*\rangle = \min_{s\in S} \langle w, s\rangle = \min_{x\in D} \langle w, x\rangle\}$.
(b) $W = \bigcup_{x^*\in Par(D)} W\,(x^*)$

A vector $w \in W$ defines a hyperplane H^w

$$H^w = \left\{ x \in \mathbb{R}^N | \langle x, w\rangle = \min_{s\in S} \langle s, w\rangle \right\}$$

Such hyperplane supports S on a subset $S^w = S \cap H^w$, and D on a face $F^w = H^w \cap D = conv\left(S^w\right)$. Both S^w and F^w belong to $Par\left(D\right)$, in particular F^w is a *Pareto face.* It is easy to verify that W is always a $\left(N-1\right)$ −polytope. For a point $s^* \in Par\left(S\right)$, the preference set $W\left(s^*\right)$ is a polytope too. In fact, the expression

$$W\left(s^*\right) = \left\{ w \in W | s^* = \min_{s \in S} \langle s, w \rangle = \min_{x \in D} \langle x, w \rangle \right\}$$

can be rewritten as

$$W\left(s^*\right) = \left\{ w \in W | \langle s^* - s, w \rangle \leq 0, s \in S \setminus \{s^*\} \right\}$$

which is equivalent to

$$W\left(s^*\right) = \left\{ w \in \mathbb{R}^N | \langle s^* - s, w \rangle \leq 0, \sum_{i=1}^{N} w_i = 1, w_i > 0, s \in S \setminus \{s^*\} \right\} \tag{2.1}$$

that is a polytope. Of course, the dimension of $W\left(s^*\right)$ cannot be more than the dimension of W. In particular $W\left(s^*\right)$ is a $(N-1)-$polytope, if there is a vector $w \in W\left(s^*\right)$ such that 2.1 holds with strict inequalities, and consequentially the hyperplane H^w supports S in $S^w = \{s^*\}$, so $F^w = conv\left(S^w\right) = \{s^*\}$. In other words, iff $W\left(s^*\right)$ has $N-1$ dimensions, then s^* is a vertex of D. A vector $\bar{w}$ such that 2.1 is defined with at least one equality, is a vector that belong to the faces of $W\left(s^*\right)$; $\bar{w}$ is preferential for a set of points $S^{\bar{w}}$ that includes s^* and as many other points, as the number of equalities. Consequentially, $F^{\bar{w}}$ is a Pareto face with one or more dimension, because it is the convex hull $conv\left(S^w\right)$ of a set that has at least two distinct points. Note that $W\left(s^*\right)$ does not necessarily contain all the faces; in fact some boundaries could be defined, according to 2.1, by one or more of the N strict inequality condition $(w_i > 0, i = 1, \ldots, N)$. The following proposition establish a relationship between the faces of $W\left(s^*\right)$ and the corresponding faces of $Par\left(D\right)$.

Proposition 2. Consider a point $s^* \in Par\left(D\right)$, and a preference vector $w \in W\left(s^*\right)$, and let $F^w = conv\left(S^w\right)$ be the corresponding Pareto face. If s^* is a vertex of D, then (a) $W\left(s^*\right)$ is a $\left(N-1\right)$ −polytope. Furthermore (b), for $(0 \leq k < N)$, if w belongs to the relative interior of a $\left(N-1-k\right)$ −face of $W\left(s^*\right)$, then the corresponding Pareto faces $F^w = conv\left(S^w\right)$ is a $k-$face.

All the reasoning done so far, plus proposition 2, lead to the conclusion that W is the union of the preference set of vertices of $Par\left(D\right)$. Such preference sets are $\left(N-1\right)$ −polytopes, and their non empty intersections are always $\left(N-1-k\right)$ −polytopes, made of vectors that are preferential for Pareto $k-$faces, $k > 0$. Notice that whenever a vector w is randomly extracted from W, then with probability 1, it will belong to the relative interior of the preference set $W\left(s^*\right)$ of some vertex s^*, then $S^w = \{s^*\}$. On top of this, several search strategies can be built.

Def. 7. Let $s^* \in Par\left(D\right)$ be a Pareto vertex; a point $s^1 \in S \setminus \{s^*\}$ is a Neighbour of s^* if it is a vertex, and if $conv\left(\left\{s^*, s^1\right\}\right)$ is a Pareto edge of D.

In other words, the neighbours of a Pareto vertex s^* are other Pareto vertices that are connected to s^* through a edge, and that simply correspond, according to Proposition 2, to the facets of polytope 2.1. Finding the facets of polytope given by the intersection of K hyperplanes is a standard computational geometry problem, (see for instance Preparata and Shamos (1993), pp.315-320 and pp.287-299). Article (Dorini et al., 2006a) shows that $Par\,(S)$ is a connected graph, meaning that it is always possible to link two vertices $s^a, s^b \in Par\,(S)$ by moving from neighbour to neighbour, through a finite sequence $s^a, s^1, s^2, \ldots, s^b$. This vertex-to-vertex approach is the idea behind Neighbour Search that enables the exploration of every vertex and edge of $Par\,(D)$ in a finite number of iterations, as shown in the following algorithm.

Algorithm 1. Neighbour Search for Convex Polytopes

Step 0 (*INITIALIZATION*). Given a random $w^0 \in W$, the first vertex s^0 can be found by solving the problem

$$\langle s, w\rangle \;\rightarrow \min_{s\in S} = \min_{x\in D} \langle x, w\rangle$$

Initialize a set $S^q = \left\{s^0\right\}$, set $S^p = \oslash$ and set $I = 0$

Step 1 if S^q is empty then go to *Step 5*, otherwise extract a point $s^* \in S^q$ and update $S^q = S^q \setminus \{s^*\}$ and $S^p = S^p \cup \{s^*\}$

Step 2 Compute the polytope

$$W\,(s^*) = \left\{ w \in \mathbb{R}^N |\, \langle s^* - s, w\rangle \leq 0, \sum_{i=1}^{N} w_i = 1, w_i > 0, s \in S \setminus \{s^*\} \right\}$$

and extract a vector for each of its M facets: $W^{facets} = \left\{w^1, w^2, \ldots, w^M\right\}$.

Step 3 If W^{facets} is empty, go to *Step 1*. Otherwise extract a vector $w \in W^{facets}$ and update $W^{facets} = W^{facets} \setminus \{w\}$.

Step 4 If H^w supports S only on two points: $S^w = \{s^*, s^v\}$, then $F^w = conv\,(S^w\}$ is a Pareto edge and s^v is a vertex. Actually this is quite always the case. In case $S^w = \left\{s^*, s^1, s^2, \ldots\right\}$, face F^w is still a Pareto edge, resulting from the convex hull of several points laying on a straight line. In order to find which point of $S^w \setminus \{s^*\}$ is the actual vertex, one can solve the following problem

$$\langle s, w\rangle \;\rightarrow \min_{s\in S^w} = \min_{x\in F^w} \langle x, \hat{w}\rangle$$

where $\hat{w}$ belongs to the set

$$\hat{W} = \{w \in W |\, \langle s^* - s, w\rangle > 0, s \in S^w \setminus \{s^*\}\}$$

Update $S^q = S^q \cup \{s^v\}$, set $I = I + 1$ and define a new set $E^I = \{s^*, s^v\}$. Finally, go to *Step 3*

Step 5 (*TERMINATION*). Set S^p contains all the vertices of $Par\,(D)$, and sets $E^1, E^2, \ldots, E^I$ generate all the I Pareto edges, i.e. $conv\left(E^i\right)$ belongs to $Par\,(D)$

Clearly the Neighbour Search is not the only way to apply proposition 2, for exploring Pareto sets. Many other approaches are possible. For example the so called *Neighbourhood Search*, described in Dorini et al. (2006a), is another algorithm based on the same principles, that can effectively find every Pareto face, and not only vertices and edges. Finally, other methodologies can be developed for performing partial searches, rather than exhaustive explorations of Pareto Sets.

2.3 Application of Neighbour Search to Markov Decision Processes

Consider a Markov Decision Process (MDP) $M = \{X, A, A(\bullet), p(\bullet), g(\bullet)\}$, where X is a finite state space; A is a finite action space; $A(x) \subset A$ are the sets of available actions at state $x \in X$; $p(y|x,a)$ are transition probabilities form $X \times A$ to X; $g(x,a) = (g_1, g_2, \ldots, g_N)$ are $N-$dimensional vectors of costs (losses), where $(x,a) \in X \times A$. A submodel M^1 of a model M, denoted with $M^1 \subseteq M$, is identical to M but with a reduced set of available actions: $A^1(x) \subseteq A(x), x \in X$. A model $M^2 \subseteq M$ is the submodel of M that is *complementary* to M^1, and it is denoted as $M^2 = M \setminus M^1$, if its action set A^2 is

$$A^2(x) = \begin{cases} A(x) & \text{if} \quad A^1(x) = A(x) \\ A(x) \setminus A^1(x) & \text{if} \quad A^1(x) \subset A(x) \end{cases}$$

A stationary randomized policy, is a transition probability $\pi(a|x)$ from X to A concentrated on the set $A(X)$. A stationary policy is *non-randomized* if π is concentrated into a single action for each $x \in X$: $\pi(\varphi(x)|x) = 1$. With some abuse of notation, φ is said to be a non-randomized policy.

According to Ionescu Tulcea theorem (Bertsekas and Shreve, 1978; Dynkin and Yushkevich, 1979), (Piunovskiy, 1997, Theorem A1.11), a policy π and an initial probability distribution μ on X define a unique probability distribution P^π_μ on the space of trajectories $(X \times A)^\infty = (x_0, a_0, x_1, \ldots)$. The corresponding mathematical expectation is denoted by E^π_μ. Notation P^π_x and E^π_x are used in case the initial distribution μ is concentrated into a single state x.

For a fixed initial distribution μ the performance of a policy π is evaluated by a vector $V^\mu(\pi) = (V^\mu_1(\pi), V^\mu_2(\pi), \ldots, V^\mu_N(\pi))$, where

$$V^\mu_i(\pi) = E^\pi_\mu\left[\sum_{t=0}^{\infty} \beta^t g_i(x_t, a_t)\right]$$

and $\beta \in (0,1)$ is the discount factor. The set D_μ of all possible vectors $V^\mu(\pi)$ under different policies π is called *Performance Set*; if the attention is restricted to non randomized policies only, the corresponding performance set is denoted with S_μ, which is a fine set of points, as the number of non randomized policies is finite. As pointed out in (Dorini et al., 2006a, Remark 4) it has been proved that D_μ is a convex polytope, whose vertices are generated by stationary non randomized policies. The vertices of a face F_μ generated by a hyperplane H, supporting D_μ, also belong to the subset $H \cap P_\mu$. Such vertices are the performances of some non randomized stationary policies, that can be combined in many ways (convex combinations), for creating other policies (Mixtures), whose performances can correspond to any point of $F_\mu = conv\,(H \cap P_\mu)$. More in general, the whole set D_μ can be generated by stationary non randomized policies: $D_\mu = conv\,(P_\mu)$. The reader can find more details on the papers (Feinberg and Shwartz, 1996; Feinberg, 2000) and on the monographs (Heyman and Sobel, 1994; Piunovskiy, 1997).

A key aspect of the applicability of the Neighbour Search to MDPs is to understand how to determine the set

$$S_\mu^w = \left\{ s \in S_\mu | \langle s, w \rangle = \min_{s \in S_\mu} \langle s, w \rangle = \min_{d \in D_\mu} \langle d, w \rangle \right\}$$

for a given vector $w \in W$. A possible way is the *Dynamic Programming* approach (DP), which is based on the following relationship:

$$\langle V^\mu (\pi), w \rangle = E_\mu^\pi \left[\sum_{t=0}^{\infty} \beta^t \langle g(x_t, a_t), w \rangle \right]$$

that makes problem $\langle d, w \rangle \to \min_{d \in D_\mu}$ equivalent to the problem

$$\langle V^\mu (\pi), w \rangle \to \min_\pi \tag{2.2}$$

For solving problem 2.2, one has to solve the Bellman equation

$$v(x) = \min_{a \in A(x)} \left\{ \langle g(x, a), w \rangle + \beta \sum_{y \in X} p(y|x, a)\, v(y) \right\} \quad x \in X \tag{2.3}$$

and then

$$\min_\pi \langle V^\mu (\pi), w \rangle = \sum_{x \in X} \mu(x)\, v(x)$$

Bellman equation can be solved using the value iteration, or, if X does not have too many elements, the policy iteration (Bertsekas, 1995). It is well known (Piunovskiy, 1997, p.53) that the minimum of equation 2.2 can be attained by any policy that belongs to the submodel M^w with the following action set:

$$A^w(x) = \left\{ a \in A(x) \,|\, v(x) = \min_{a \in A(x)} \left\{ \langle g(x,a), w \rangle + \beta \sum_{y \in X} p(y|x,a)\, v(y) \right\} \right\} \quad x \in X \tag{2.4}$$

Set S_μ^w is the collections of all the performances generated by all non randomized policies of M^w; similarily, $F_\mu^w = conv\left(S_\mu^w\right) = D_\mu^w$: the face F_μ^w of D_μ^w coincised with the total performance set of the submodel M^w. It is very important to notice that M^w does not depend on the initial distribution μ. In order to obtain the coordinates of the points in S_μ^w, one has to evaluate every policy φ from M^w, solving the equation

$$J_i(\varphi, x) = g_i(x, \varphi(x)) + \beta \sum_{y \in X} p(y|x, \varphi(x))\, J_i(\varphi, y) \quad , x \in X, i \in \{1, \ldots, K\} \tag{2.5}$$

so that $V_i^\mu(\varphi) = \sum_{x \in X} \mu(x)\, J_i(\varphi, x)$.

A second key aspect of the applicability of the Neighbour Search to MDPs is to understand how to determine the preference set of a vertex. As already mentioned, for a vector $\hat{w}$ that is randomly selected from W, it is theoretically garanteed and practically safe to assume, that $S^{\hat{w}}$ only contains a Pareto vertex s^*; that means, $V^\mu(\pi) = s^*$ for every π of $M^{\hat{w}}$. In order to define the preference set $W(s^*)$, consider a fix non randomized policy $\hat{\varphi}$ from $M^{\hat{w}}$: $\hat{\varphi}(x) \in A^{\hat{w}}(x)$ for all $x \in X$.

$$W(s^*) = \left\{ w \in W | s^* = \min_{s \in S_\mu} \langle s, w \rangle = \min_{x \in D_\mu} \langle x, w \rangle \right\}$$

is equivalent to the following

$$W(s^*) = \{ w \in W | \hat{\varphi}(x) \in A^w(x), x \in X \} \tag{2.6}$$

Where A^w is defined by 2.4. Clearly the preference set is not useful in the form 2.6, and it has to be turn into an explicit intersection of halfspaces like 2.1. The equation 2.3 is identical to

$$v(x) = \min_{a \in A(x)} \left\{ \langle g(x,a), w \rangle + \beta \sum_{y \in X} p(y|x, \hat{\varphi}(x))\, v(y) \right\} \quad x \in X, w \in W(s^*) \tag{2.7}$$

and it is easy to verify that

$$v(x) = \langle J(\hat{\varphi}, x), w \rangle, x \in X \tag{2.8}$$

substituting 2.8 into 2.7 results in

$$\langle J(\hat{\varphi}, x), w \rangle = \langle Q(\hat{\varphi}, x, \hat{\varphi}(x)), w \rangle = \min_{a \in A(x)} \langle Q(\hat{\varphi}, x, a), w \rangle \quad x \in X, w \in W(s^*) \tag{2.9}$$

where

$$Q_i(\hat{\varphi}, x, a) = g_i(x, a) + \beta \sum_{y \in X} p(y|x, \hat{\varphi}(x)) J_i(\hat{\varphi}, y) \quad , a \in A(x), x \in X \tag{2.10}$$

is called *Q-function*, and it only depends on $\hat{\varphi}$. At this point, the set A^w can be redefined with respect of the Q-function:

$$A^w(x) = \left\{ a \in A(x) \,|\, \langle Q(\hat{\varphi}, x, \hat{\varphi}(x)), w \rangle = \min_{a \in A(x)} \langle Q(\hat{\varphi}, x, a), w \rangle \right\} \ x \in X, w \in W(s^*) \tag{2.11}$$

or, equivalently

$$A^w(x) = \{ a \in A(x) \,|\, \langle Q(\hat{\varphi}, x, \hat{\varphi}(x)) - Q(\hat{\varphi}, x, a), w \rangle \leq 0 \} \ x \in X, w \in W(s^*) \tag{2.12}$$

finally the preference set 2.6 can be redefined as

$$W(s^*) = \{ w \in W | \langle Q(\hat{\varphi}, x, \hat{\varphi}(x)) - Q(\hat{\varphi}, x, a), w \rangle \leq 0 \} \ a \in A(x), x \in X \tag{2.13}$$

which is the intersection of halfspaces that are exclusively defined by the Q-function. For a vector w that belongs to the relative interior of $W(s^*)$, the set A^w resulting from 2.12, always coincides with $A^{\hat{w}}$. If w belongs to a face of $W(s^*)$, then, for some $x \in X$, A^w will be more rich: $A^{\hat{w}}(x) \subseteq A^w(x)$, thus $M^{\hat{w}} \subset M^w$. The performance set S^w_μ generates a face D^w_μ given by the convex combinations of non randomized policies in $M^{\hat{w}}$ with non randomized polices of $M^w \setminus M^{\hat{w}}$. In particular, if w belongs to a facet of $W(s^*)$, then D^w_μ is a edge, and most likely, all the policies of $M^w \setminus M^{\hat{w}}$, generate the other edge, hence, $P^w_\mu = \{s^*, s^v\}$, (see Algorithm 1, *Step 4*). However, in the general case, submodel $M^w \setminus M^{\hat{w}}$ could contain non randomized policies, resulting in several distinct points laying on a straight line. In such case, s^v and the corresponding submodel should be determined. A possible action scheme follows:

a) extract a non randomized policy φ from $M^w \setminus M^{\hat{w}}$, and calculate the function $J_i(\varphi, x), x \in X, i \in \{1, \ldots, K\}$

b) Denoting with $\tilde{A}$ the action set of $M^w \setminus M^{\hat{w}}$, verify that for every $a \in \tilde{A}(x)$

$$J_i(\varphi, x) = g_i(x, a) + \beta \sum_{y \in X} p(y|x, \varphi(x)) J_i(\varphi, y) \quad , x \in X, i \in \{1, \ldots, K\}$$

if that is the case, every policy of the complementary model generate a single point that is the other vertex of the edge: $s^v_i = V^\mu_i(\varphi) = \sum_{x \in X} \mu(x) J_i(\varphi, x)$.

c) If statement in point b) is not verified, the model that generates s^v can be found by solving the Bellman equation again; the resulting action set is the following

$$\left\{ a \in \tilde{A}\left(x\right) | v\left(x\right) = \min_{a \in \tilde{A}(x)} \left\langle g\left(x,a\right), w\right\rangle + \beta \sum_{y \in X} p\left(y|x,a\right) v\left(y\right) \right\} \; x \in X$$

where w must belong to a set

$$W\left(s^{*}\right) = \left\{ w \in W | \left\langle Q\left(\varphi, x, \varphi\left(x\right)\right) - Q\left(\varphi, x, a\right), w\right\rangle > 0 \right\} \; a \in \tilde{A}\left(x\right), x \in X.$$

Introducing the scheme above, and definition 2.13, in the algorithm 1, is possible to explore the Pareto set of a Markov Decision Process through the Neighbor Search approach.

Chapter 3

The case study: Hoa Binh system

3.1 The System

Hoa Binh is the largest reservoir in Vietnam, with a total storage capacity of 9.5 billion m^3 and a live storage of 5.6 billion m^3. It has been operated in the Red river basin since 1990, its two main purposes being flood control and hydropower generation.

3.1.1 Red river basin

Red river has a total catchment area of 169,000 km^2, 50% of it lies in Vietnam, the remainder in China and Laos (see figure 3.1).

Upstream of Hanoi the three major tributaries *Da*, *Thao* and *Lo* join to form the Red river delta. This region, about 17,000 km^2 of flat land only 2 m above sea level, is actually a high density (1,000 persons per km^2) mainly rural area. Delta region is now fundamental for Vietnam agriculture and, according to government forecasts, in the next decades it will incur a strong urbanization and industrialization. The delta (including Hanoi) has already a population of 17 million people, 70% of the whole Red river basin population (Tinh, 1999).

Another key aspect of the system is the basin climate and its hydrological characteristics: mainly in a subtropical area, basin climate is dominated by the monsoon winds of East Asia. This implies a strong seasonal component in rainfall distribution. The mean annual rainfall is in the range 1200–4800 mm, but only 20% of it falls in the dry season, from November to April. The remainders falls in the rainy season, from May to October. Consequently the flow of the Red river varies during the year.[1]

Economical and social importance, along with the vulnerability of the area (an average of 6

[1] from a minimum recorded discharge of 370 m^3/s to a peak of 38,000 m^3/s measured at Hanoi during the flood event of 1971.

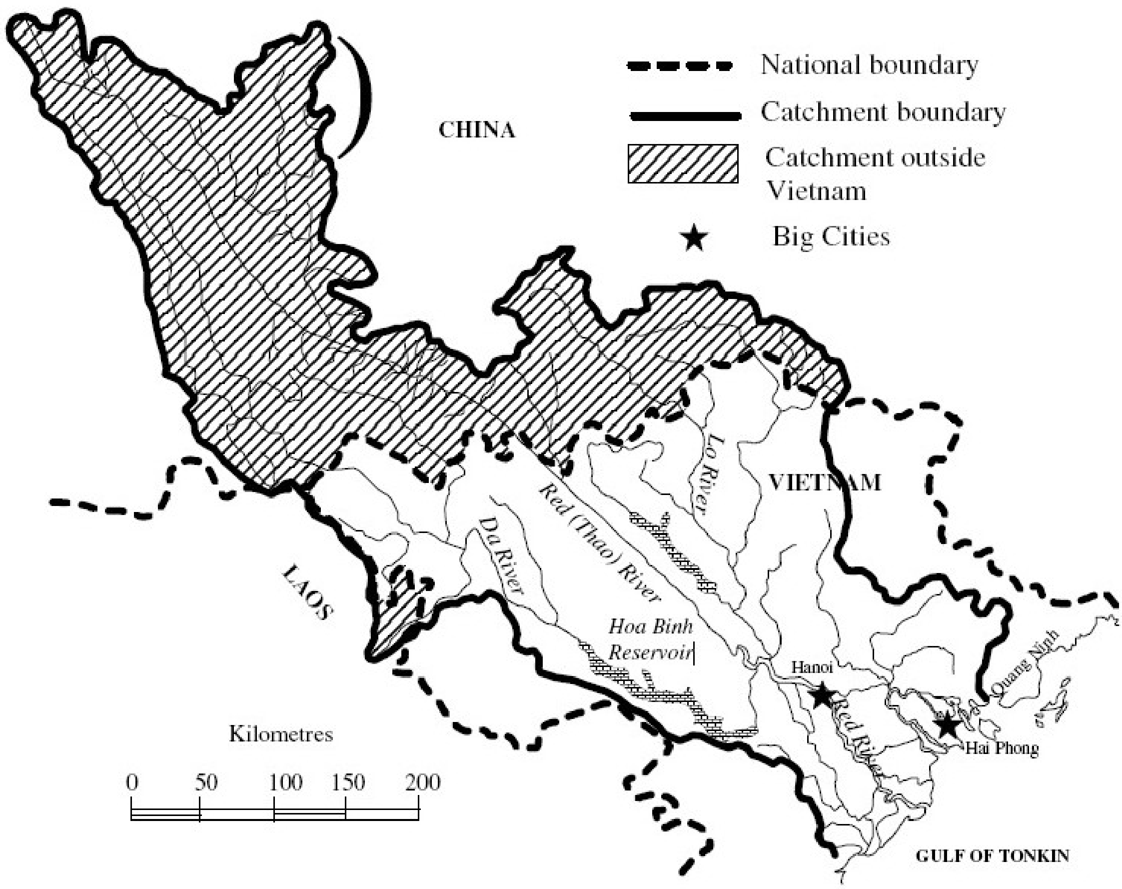


Figure 3.1: Map of the Red River basin (Le Ngo et al., 2006)

typhoons a year hit the coastal area) explain the need of the population for a flood control system, which is designed around the Hoa Binh dam and its operation.

3.1.2 Hoa Binh dam

The Hoa Binh reservoir drains the water of the main Red river tributary (see table 3.1), the *Da* river.

The dam was designed to reduce the peak flood level at Hanoi by 1.5 m during flood events with a return period of 200 years (like the one occurred in 1971). New assessments estimate that human intervention affected the riverbed, reducing the reservoir effect to only 0.6 m Tinh (1999).

Hydroelectric power plant connected to the reservoir is equipped with eight 240 MW turbines, corresponding to a maximum power generation capacity of 1920 MW. The annual average production

	Da	*Thao*	*Lo*
μ	3268.8	1968.1	2139.6
σ	2111.6	1282.8	1269.4

Table 3.1: Average and standard deviation of flows (m^3/s) in the three tributaries over 20 rain seasons (between 1963 and 1996).

is 7.8 billion kWh, approximately 40% of the whole Vietnam electric supply.

Due to its great importance for the country energy resources, maximizing hydropower production is the second objective of Hoa Binh operation. Actually during the dry season it is the main one, and operation of the reservoir is directly controlled by Electricity Viet Nam. Only in the period from 15 June to 15 September, the control passes to the Central Commitee for Flood Control.

The problem faced in this work concerns these three months of the year only, when hydropower maximization conflicts with flood control purposes so, in what follows, only this period will be considered.

Additionally to turbines, the reservoir has several discharge ways that can be controlled to raise or lower releases. The controller may activate 12 bottom gates and 6 spillways to increase outflow (Le Ngo et al., 2006). Whereas turbines admit continuous control (i.e. each release between 0 and 2400 m^3/s is allowed), bottom gates and spillways can be only completely open or close. Obviously spillways are not effective if water level in the reservoir is below a certain threshold (+102.5 m, corresponding to 7.04 billion m^3 of water stored).
The only rule that must be respected in releases control is that the first six bottom gates that are opened or the last six that are closed have to be operated with 6 hours of gap between each operation.[2] The purpose of this rule is to avoid too fast variation of downstream flow, considering that discharge through two bottom gates (1000 - 1800 m^3/s each) is of the same magnitude of the discharge through the eight turbines combined (2400 m^3/s). However, if the six first bottom gates are opened, any operation of further bottom gates and spillways is allowed.

3.1.3 Actual regulation

Operations rule actually in use for the Hoa Binh reservoir during the flood season is based on four main key parameters (Le Ngo et al., 2006; CCFSC, 2005):

Water level at Hanoi. Because Hanoi is the most important site for flood control in the Red river basin, the water level at Hanoi is a key parameter to measure the safety level of the flood control system. It is also a representative characteristic for the dyke system in the basin.

Water level at Hoa Binh reservoir. As just explained, Hoa Binh reservoir plays an important

[2] In what follows this operating rule will be addressed as the “6 hours rule”.

role in flood control in the basin. Keeping water level low allows to store major floods but threatens hydropower supply.

Hydrological forecast information. One of the most important inputs for actual reservoir operation is hydrological forecast information. In this case 24-hour forecasts of the reservoir inflow and of the water level at Hanoi are used in the regulation. Data used to perform forecasts are flows on the Da, Thao and Lo rivers measured in upstream stations[3], and outflow from the Hoa Binh reservoir.

Season. In order to ensure both flood protection and efficient hydropower generation, three regulation periods have been defined:

- Pre-flood season from 15 of June to 15 of July;
- Main flood season from 16 of July to 20 of August;
- Post-flood season from 21 of August to 15 September.

Target water levels and other parameters of the operation rule vary from period to period.

Operating rule is based on a strict hierarchy of objectives: reservoir protection is the primary one, flood control the second and at last hydropower generation. On this basis, rule results in a sequence of evaluation of the key parameters above, that lead to an opportune operational procedure.
For example, the procedure for reducing regular floods is applied if predicted level at Hanoi exceed +11.50 m within the next 24 hours and level in the Hoa Binh reservoir is below +100 m. It consists in reducing reservoir release closing the turbines; the aim is to keep level at Hanoi below +11.50 while avoiding that the water level in the reservoir exceeds +100 m.
If level at Hanoi is expected to be below the flood threshold, the operational procedures applied are only power generation-aimed. Otherwise, if a major flood is occurring, higher levels are admitted in the reservoir to smooth the flood event. If the stored water level reaches +120 m the priority becomes reservoir protection and the release is then kept as much similar as possible to the inflows, to avoid a further level raising.

This hierarchic control system is presented in Le Ngo et al. (2006), where it is modeled and described as a decision tree. The tree is constituted by a list of more than 100 logical statements referred to the key parameters and ordered according to the priorities given.

The operating rule optimization proposed in Madsen et al. (2006) is based on this rule model. The optimization is made varying seven thresholds that identifies the different operational procedures. I.e., in the previous example the +100 m level that limits the regular floods procedure is moved in a feasible parameter space [+100 m, +103 m] to find the optimum. The objectives optimized are pertaining to flood reduction and hydropower production (the objectives are formally stated in section 3.4).

[3] In the order: *Ta Bu*, *Phu Tho* and *Vu Quang* measurement stations.

3.1.4 Available data

For this research, the data used belongs to two classes:

Measured data. Direct measures for 20 flood seasons in the period between 1963 and 1996 are available. The time interval between each measure is 6 hours.

The measured data used are flows in the three tributaries Da, Thao and Lo.

Calculated data. Measures cover mainly a period during which the dam was not operating yet, or not even built. Consequently, for the purpose of optimizing the reservoir operations, direct measures of downstream variables are useless. On the other hand a hydraulic description of the system (including the reservoir) has been implemented in the MIKE-11 model (Le Ngo et al., 2006; Madsen et al., 2006; DHI, 2005). Thanks to this model high-quality hydraulic simulation of the system are available for the 20 seasons of measure as if the reservoir was already operating, obtaining estimations of the downstream variables of interest. Also this data are available with time intervals of 6 hours.

The calculated data employed are reservoir water level and release, and water level in Hanoi.

The further information available about the system are the release functions for bottom gates and spillways and the production curve for turbines, expressing hydropower generation in relation to headwater and flow through turbines. The release functions are given by several (71) points, the production curve by an analytical interpolation function. Both are described in section 3.2.

3.2 The Models

In this section the modeling of the Hoa Binh system is detailed. A logical scheme of such system is showed in figure 3.2 (Madsen et al., 2006; Le Ngo et al., 2006).

Beside the *planning model* that is used in the optimization process, two more models are considered here:

The Validation Model. In this work a comparison term is offered by the studies of Le Ngo et al. (2006) and Madsen et al. (2006). Also if the aims are different, and consequently the results can not be directly compared, their work is used as a validation test. With this purpose a second model is prepared to verify some of the assumptions made[4].

The POLFC Model. Once obtained an operating rule for the planning model, it could be applied to the real system through a POLFC (Partial Open Loop Feedback Control) in order to improve the policy effectiveness (Bertsekas, 1995). As the purpose of this work is to test the

[4]In particular about the discretization described in section 3.3 and about the validity of the control law optimized. The comparison is discussed in section 3.4.

NS algorithm, and the planning problem is already suitable for this, the POLFC application is not implemented. However, it is shortly discussed in section 3.2.7.

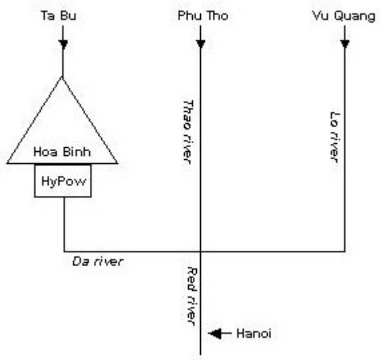


Figure 3.2: Scheme of the system

3.2.1 Definitions

The variables used in what follows are:

- $w_t^{Da}, w_t^{Thao}, w_t^{Lo} \in \mathbb{R}^+$, average flows (in m^3/s) in the three tributaries during the time step $[t, t+1)$.
- $b_t \in X_1$, number of dam's bottom gates opened at time t.
- $s_t \in X_2$, reservoir's storage (in billion m^3) at time t.
- $h_t \in X_3$, maximum water-level at Hanoi registered during the time-step $(t-1, t]$.
- $a_t \in A$, regulation of dam's release during the time step $[t, t+1)$.
- $r_t \in \mathbb{R}^+$, average release from the reservoir (in m^3/s) during the time step $[t, t+1)$.
- $\Delta t \in \mathbb{R}^+$, duration (in hours) of the time step $[t, t+1)$.

3.2.2 Time-steps

Different time-steps have to be considered in the model setting: the main constraint for the choice of the minimal pace come from the available measures, which have a 6-hours interval between each. The control time-step can also be reasonably considered 6-hours long, remembering the "6-hours rule" described in section 3.1.3 that limits application of more frequent decisions.

On the other hand, a 6-hours time-step could not be practical (nor really useful) for a model with only planning purpose. In fact, managing a system with 6-hours time-steps for seasons of 4 months could only strongly increase the state dimension without bringing effective improvements. Outside major flood events, actually, a decision-making with frequency higher than daily seems normally not necessary. The implementation of the POLFC will anyway give a more detailed control when it will be necessary.

Consequently, for the planning and validation model a 48-hours time-step is used, with decisions made of a sequence of 8 elementary controls (see section 3.2.4). This is both to keep models state-dimension low, and to make easy to integrate, for example, a 6-hours time-step POLFC with the policies obtained.

3.2.3 The upstream model

Inflows to the system from the three tributaries are $w_t^{Da}, w_t^{Thao}, w_t^{Lo}$. Their description changes in the three models and it actually represents the main difference between them. The POLFC model description must be the most accurate. The two other will be simpler.

$w_t^{Da}, w_t^{Thao}, w_t^{Lo}$ are expressed in m^3/s and represent the average inflows from the three rivers, assumed equally distributed throughout $[t, t+1)$. In all the three models, they are treated as stochastic variables:

$$w_t^{Da} \sim \phi_t^{Da}(\cdot),\ w_t^{Thao} \sim \phi_t^{Thao}(\cdot),\ w_t^{Lo} \sim \phi_t^{Lo}(\cdot)$$

The description of inflows for the POLFC model is not detailed in this thesis, but in section 3.2.7 its form is discussed.

For the other two models purely stochastic variables are used, lognormal distributed:

$$\log w_t^{Da} \sim N(\mu^{Da}, \sigma^{Da}),\ \log w_t^{Thao} \sim N(\mu^{Thao}, \sigma^{Thao}),\ \log w_t^{Lo} \sim N(\mu^{Lo}, \sigma^{Lo}) \tag{3.1}$$

The parameters of the distributions are different for the planning and validation model:

Planning model. To calibrate the inflows model all the available time-series (20 years) have been used. Through the analysis of these series, however, it has been decided to consider only the central period of each (i.e.: from 1 July to 2 September). This period is the core of the flood season, when rainfall is more abundant and water management is more delicate. In figure 3.2.3 average and standard variation for the Da river flows during the season are shown; the period considered is indicated by the vertical lines.

Validation model. The purpose of this model being the comparison with Madsen et al. (2006), it has been calibrated using the synthetic time-series presented in the cited article. This series is made of four flood-season records: the 1971 series (i.e. the inflows that generated the most severe flooding recorded) and three synthetically generated seasons. Such series have been generated scaling existing records in order to obtain an Hanoi flooding of the same level of the 1971's.

This operation results in a inflows scenario more severe than the historical one and more challenging for flooding prevention policies. Moreover, some optimization result is available for this scenario. These series are, consequently, a good benchmark for the model and the consequent MDP described here.

The distribution parameters for the two models are listed in table 3.2

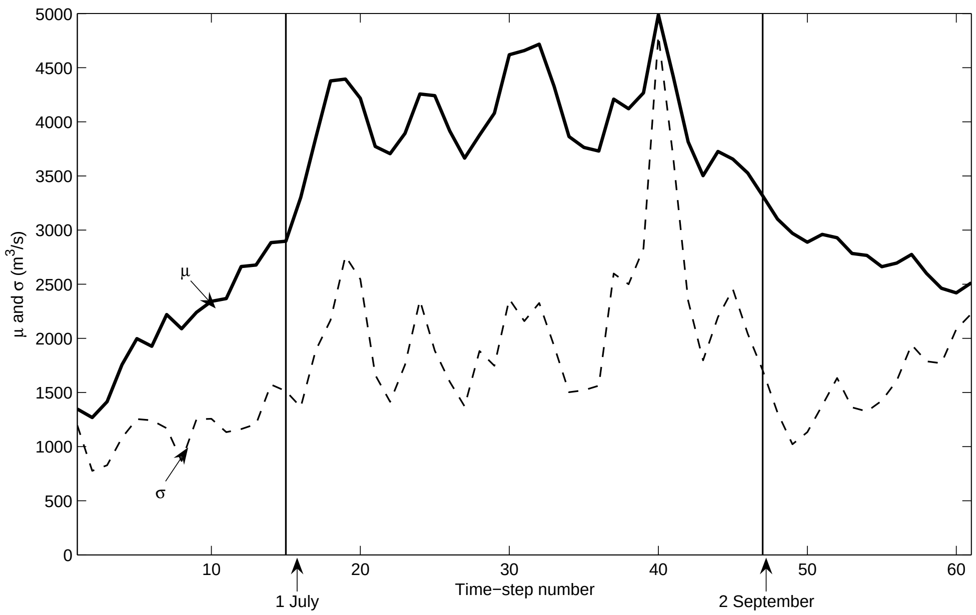


Figure 3.3: Average and standard deviation of the Da river flow during the rain season

Model	μ^{Da}	σ^{Da}	μ^{Thao}	σ^{Thao}	μ^{Lo}	σ^{Lo}
Planning	8.1609	0.51357	7.5772	0.54717	7.7071	0.4857
Validation	8.5649	0.4709	7.9644	0.5756	8.0523	0.4941

Table 3.2: Parameter set for inflows distribution

This simple description of the inflows presents two main issues:

- a purely stochastic model does not take into account the inflow self-correlations typical of a large basin,

- a stationary model does not represent the seasonal variations of rainfall.

A classical solution to these issues is, for example, the use of ARMA models as suggested by Lamond and Boukhtouta (2002). However in this work the first issue has been indirectly faced introducing an auto-regressive component in the downstream model (i.e.: the model that estimates Hanoi water-level, detailed in section 3.2.5). Further, the role of taking into account any self-correlation and time-dependence of the inflows is left to the POLFC, which can deal with it without surcharging the planning phase.

3.2.4 The reservoir

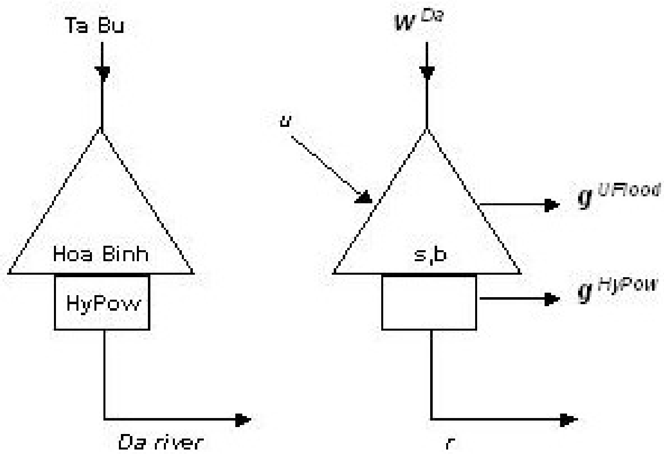


Figure 3.4: Reservoir model scheme

The storage

The reservoir storage (s) can vary between 4.4 and 10.45 Gm^3, and its evolution is described by the mass-balance differential equation:

$$\frac{ds(t,s,b,a)}{dt} = \bar{i}_t(s) - \bar{r}_t(s,b,a),\ \ s \in X_2 = [4.4, 10.45]\,, \tag{3.2}$$

where $\bar{i}_t(s)$ is the net inflow in the reservoir at time t (expressed in m^3/s). It can be generally expressed as

$$\bar{i}_t(s) = \overline{w}_t^{Da} - e_t(s) + q_t(s)$$

where $e_t(s)$ is the instant evaporation from the reservoir depending on the water surface. It is assumed that this term can be disregarded during the flood season, when air moisture is generally high. $q_t(s)$ is the rainfall over the reservoir. Considering the surfaces of the basin producing the inflows and of the reservoir, also this term is disregarded. Consequently equation 3.2 can be written:

$$\frac{ds(t,s,b,a)}{dt} = \overline{w}_t^{Da} - \overline{r}_t(s,b,a) \tag{3.3}$$

This equation can be integrated on a time interval $[t, t+\tau]$:

$$s_{t+\tau}(s,b,a) = s_t + (w_t^{Da} - r_t(s,b,a)) \cdot \tau \tag{3.4}$$

where w_t^{Da} and $r_t(s,b,a)$ are, respectively, the average instant inflow from the Da river and the average instant release from the reservoir in the τ interval.
As described in section 3.2.2, the control could change each 6 hours. On the other hand, a such time-step is too long to accurately describe dependence of outflow from storage s. Then the choice is $\tau = 1200$ s, after a sensitivity analysis: the release variation have been measured from the reference case of 1-second time-steps both in a fast-filling situation and in a fast-emptying one for several τ values. The 20-minutes pace performed as a good trade-off between precision (difference from 1s-case < 0.1 %) and number of iterations per time-step (18 per 6-hours step, 144 per 48-hours step).

Finally, the $r_t(s,b,a)$ term in equation 3.4 is a linear interpolation of the instant release curve (in what follows shortly called release curve) considered constant over a 20-minutes interval[5].

The release curve

The release from the reservoir is the sum of several terms, corresponding to turbines, bottom gates, spillways and extra-spillways.

The turbines can release from 0 to 300 m^3/s each, corresponding to a total range, completely controllable, from 0 to 2400 m^3/s.

The release curves for spillways and bottom gates that, as stated, can be only completely closed or opened, have been interpolated from disposable data points (see figure 3.5). Two 4th-order polynomial curves have been used, with a R^2 value of 0.9996 and 0.998 respectively for bottom gates and spillways.

In dam description there is no automatic spillway for over-storing events. This probably because the main priority of the actual regulation (see section 3.1.3) is to prevent such events to protect dam. However in the present model such priority is not given; consequently a upper limit to storage capacity must be set. This limit has been implemented as an additional automatic release that

[5] Also a quadratic interpolation for the release curve has been experimented. Actually, noise in the points defining the curve was too high to obtain a smooth second derivative of the curve itself, and it was more convenient to stop at the first order.

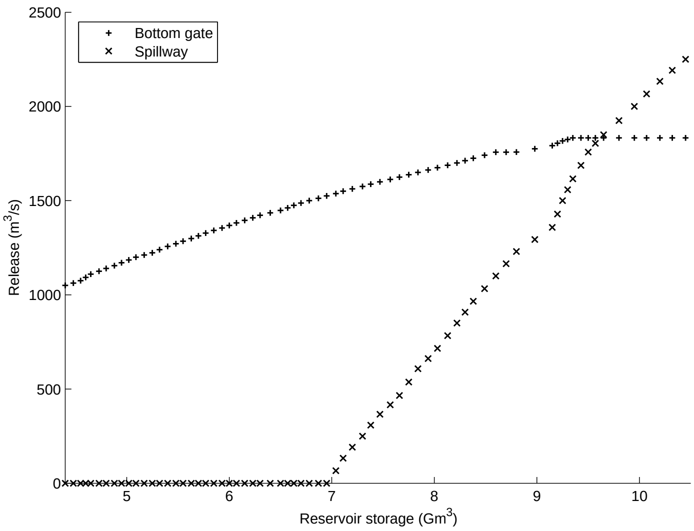


Figure 3.5: Release curve for each bottom gate and spillway

prevents the reservoir to store more than the maximal capacity (i.e.: 10.45 billion m^3). To describe the dam-protection priority in the model, a further cost is added, the *upstream flooding*, that is presented in section 3.2.6.

The control system

Description of controls as a simple release decision is not straightforward for the Hoa Binh reservoir. As mentioned, in facts, many release-way are available, different time-steps should be used in the models and the 6-hours rule limits the operations.
Some hypothesis are made on the control system:

- There is no point in keeping any bottom gate or spillways open if turbines are closed. In other words, until the maximum turbines outflow capacity is reached, any further release must be done through turbines;
- There is no difference in releasing a certain flow through bottom gates or spillways.
- Each of the eight turbines can be only opened or closed. This corresponds to a 9-classes discretization of the release range [0,2400] m^3/s.

Controls are implemented in terms of "release-ways opened at the end of the time-step". This means that given a control as *"all turbines and bottom gates opened, one spillway open"*, the system will react opening or closing any release-way necessary to reach, *as soon as possible*, the desired configuration, and then keeping it until next decision is taken.

"As soon as possible" is to underline that not any control can be immediately executed. Actually, if time-step is too short, some controls can not be completed before another decision is taken[6].

On the other hand, this description allows all possible regulations only for a 6-hours time-step. For a longer one some possible regulation is lost (e.g.: the alternate opening and closing of a bottom gate each 6 hours) but a reasonable set of it is maintained. This assumption is part of the validation discussed in section 3.4.

The obtained control set A is defined:

$$A = \{1, \ldots, 27\} \tag{3.5}$$

The meaning of each decision is described by:

$$\begin{aligned}
a_t = 1 \quad &\rightarrow \quad \text{no release-way opened at time } t+1,\\
a_t = i,\ i \in [2, \ldots, 9] \quad &\rightarrow \quad i-1 \text{ turbines opened at time } t+1,\\
a_t = i,\ i \in [10, \ldots, 21] \quad &\rightarrow \quad 8 \text{ turbines and } i-9 \text{ bottom gates}\\
& \quad \text{opened at time } t+1,\\
a_t = i,\ i \in [22, \ldots, 27] \quad &\rightarrow \quad 8 \text{ turbines, } 12 \text{ bottom gates and}\\
& \quad i-21 \text{ spillways opened at time } t+1,
\end{aligned} \tag{3.6}$$

To implement the 6-hours rule, and so to describe bottom gates dynamics in the model, the dimension b is added to the state of the system, defined as follow:

$$\begin{aligned}
b_t = 1 \quad &\rightarrow \quad \text{no bottom gates are opened at time } t,\\
b_t = i,\ i \in [2, \ldots, 6] \quad &\rightarrow \quad i-1 \text{ bottom gates are opened at time } t,\\
b_t = 7 \quad &\rightarrow \quad 6 \text{ or more bottom gates are opened at time } t.
\end{aligned}$$

[6] As an example, it can be considered a 24-hours time-step: if an "all bottom gates opened" control is given and all bottom gates are closed, the control execution will require 30 hours (6 hours for each bottom gate, the first opened immediately). Before this time passed, another decision is taken.

The admitted control set A_t can be consequently defined as $A(b_t)$ [7], and the b-state evolution is described by the deterministic equation:

$$b_{t+1} = \begin{cases} 1 & \text{if } a_t \leq 9 \\ a_t - 8 & \text{if } 9 < a_t < 15 \\ 7 & \text{otherwise} \end{cases} \quad , a_t \in A(b_t) \tag{3.7}$$

The hydropower plant

Let us define

$$\begin{aligned} h_t^{Upstream} &= 46.758 \cdot s_t^{0.402} \\ h_t^{Downstream} &= 8 \cdot 10^{-12} \cdot r_t^3 - 2 \cdot 10^{-7} \cdot r_t^2 + 2.3 \cdot 10^{-3} \cdot r_t + 10.936 \\ \Delta h_t &= h_t^{Upstream} - h_t^{Downstream} \end{aligned} \tag{3.8}$$

where $h_t^{Upstream}, h_t^{Downstream}$ are the water levels, respectively upstream and downstream[8], and consequently Δh_t is the headwater.

Let $r_t^{Turbines} \in [0, 2400]$ the average release through turbines (m^3/s) during $[t, t+1)$, and $P_t \in [0, 1920]$ the average hydroelectric production (MW) during $[t, t+1)$. $r_t^{Turbines}$ is defined by:

$$r_t^{Turbines} = \min(r_t, 2400) \tag{3.9}$$

The power generation is described by the function:

$$P_t = \begin{cases} \frac{r_t^{Turbines} \cdot 3.6}{10.56 - 0.069 \cdot \Delta h_t} & \text{if } \Delta h_t < 88\text{m}, \\ \\ \frac{r_t^{Turbines} \cdot 3.6}{7.12 - 0.03 \cdot \Delta h_t} & \text{if } \Delta h_t >= 88\text{m}. \end{cases} \tag{3.10}$$

As described in section 3.2.4, the reservoir integration time-step τ is different from Δt, and so $n = t/\tau$ values $s_{t+k\tau}$ and $r_{t+k\tau}$, $k \in [1, \ldots, n]$ are available. Consequently a time-aggregation

[7] For $[t, t+1) = 48$h, $A(b_t) \equiv A$ because each control can be applied in the timestep, while for $\Delta t = 6$h, $A(b_t)$ is defined as follow:

$$\begin{aligned} A(1) &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}, \\ A(2) &= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}, \\ A(3) &= \{11, 12, 13\}, \\ A(4) &= \{12, 13, 14\}, \\ A(5) &= \{14, 15, 16\}, \\ A(6) &= \{15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27\}, \\ A(7) &= \{16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27\}. \end{aligned}$$

[8] The two functions are interpolation from the data set; respectively R^2=0.9996 and R^2=0.9917

criterion for $P_{t+k\tau}$ must be selected:

$$P_t = \frac{1}{n}\sum_{i=1}^{n} P_{t+i\tau} \tag{3.11}$$

3.2.5 The downstream model

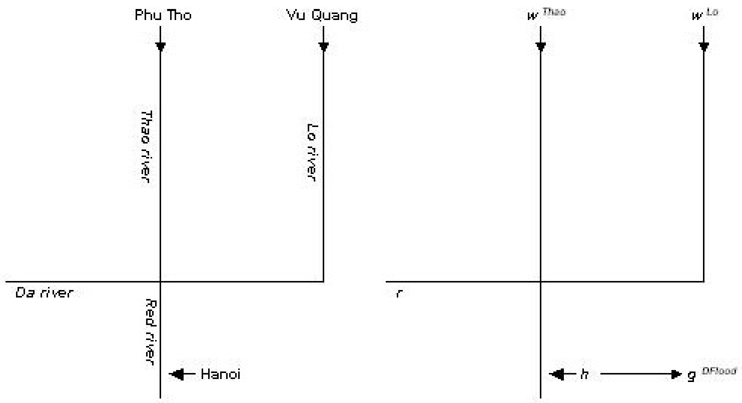


Figure 3.6: Downstream model scheme

The downstream model purpose is to evaluate the water level at Hanoi, through the relation:

$$h_t = f(r_t,\ w_t^{Thao},\ w_t^{Lo}) \tag{3.12}$$

As mentioned in section 3.2.3, to take into account auto-regressive dependence of inflows, an auto-regressive term is introduced in the downstream model. So h is treated as a state variable, described by equation 3.12 that becomes:

$$h_t = f(r_t,\ w_t^{Thao},\ w_t^{Lo},\ h_{t-1}) \tag{3.13}$$

Further, the data set available for $\Delta t = 48$h is larger than r_t, w_t^{Thao}, w_t^{Lo}, h_{t-1}. In fact the reservoir's equation integration provides the series:

$$\{r_t, r_{t+\tau}, r_{t+2\tau}, \ldots, r_{t+7\tau}\},$$

with $\tau = 6$h. Consequently, the h state evolution can be better described by:

$$h_t = f(r_t, \ldots, r_{t+7\tau},\ w_t^{Thao},\ w_t^{Lo},\ h_{t-1}), \quad \text{if } \Delta t = 48\text{h} \tag{3.14}$$

To estimate this relation from the time series an artificial neural network has been implemented (in what follows: ANN). A feed-forward network made by two layers of neurons has been used.

The calibration series is made by 7000 inputs structured as follow: 8 6-hours average releases, *Thao* and *Lo* 48-hours average inflows and Hanoi maximum level in the past 48-hours. A series of 2712 inputs has been used for validation.

Good results are obtained (see figure 3.7) with an hidden layer of 10 neurons with a tan-sigmoid transfer function; followed by a linear, single-neuron output layer.

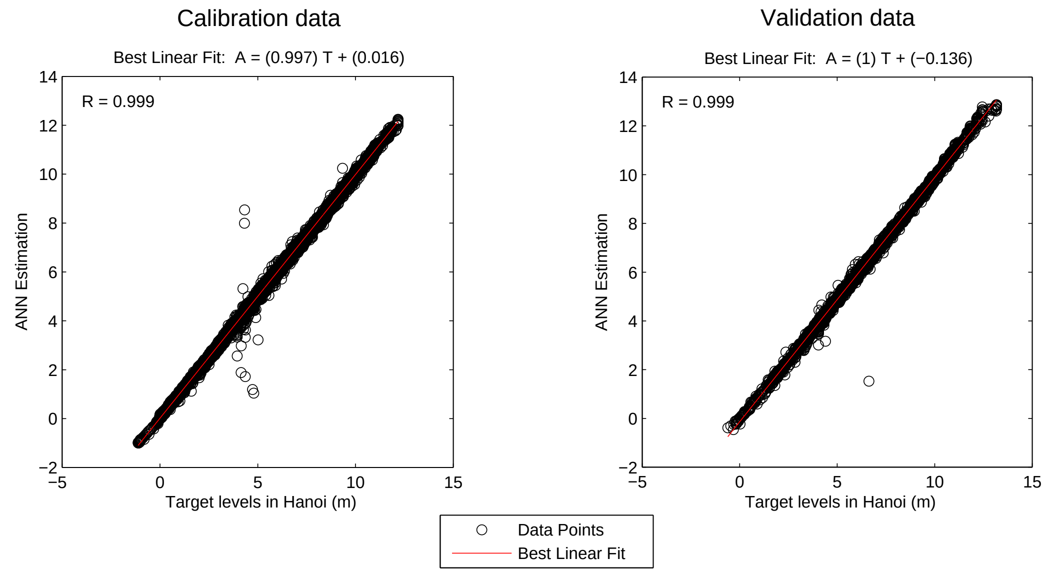


Figure 3.7: Neural network calibration results

3.2.6 Costs-per-stage

Three cost-per-stage function are defined.

Downstream Flood. It represents the Hanoi-flooding cost.

Hydropower Generation. It is the hydroelectric power plant unemployed capacity during the time-step $[t, t + 1]$.

Upstream Flood. This cost shows if the water level approaches or exceed the reservoir critical storage capacity.

For the planning model the costs-per-stage are defined:

$$
\begin{aligned}
g_t^{DFlood} &= \begin{cases} 0 & \text{if } h_t \leq 11.5 \text{ m} \\ (h_t - 11.5)^3 & \text{else} \end{cases} \\
g_t^{HyPow} &= \begin{cases} 0 & \text{if } P_t = 1920 \text{ MW} \\ 1920 - P_t & \text{else} \end{cases} \\
g_t^{UFlood} &= \begin{cases} 0 & \text{if } s_t \leq 9 \cdot 10^9 \text{ m}^3 \\ (s_t - 9)^5 & \text{else} \end{cases}
\end{aligned}
\tag{3.15}
$$

3.2.7 Partial Open-loop Feedback Control

As mentioned in the beginning of this chapter, the problem setting described is intended for the Hoa Binh reservoir operations *planning*.

So, optimal controls resulting from that optimization will not be applied directly on the real system. However they could be used, through the corresponding optimal cost-to-go function, in an on-line control system that will provide the actual regulation to apply to the reservoir.

This approach has positive effects both on the planning process, and on the system actual management. On the planning side, simplifications can be done without relevant losses on final performances; consequently the costs linked to optimization (time, resources) are reduced. On the management side it is possible, thanks to on-line control, to involve in the decision a set of informations about the system not available for the planning process. A typical example of such informations for reservoir operation are weather forecasts over the basin.

One effective on-line control system is the Partial Open-loop Feedback Control, discussed by Bertsekas and Tsitsiklis (1996). Using a POLFC consists in determining the control to apply at time t through a finite-horizon optimization computed at the same time t. Such optimization is connected to a previous less detailed planning, for example on a infinite horizon, through the optimal cost-to-go function: the final step of each POLFC optimization must coincide with a step of the planning; the cost-to-go function of the planning is used as a final cost for the POLFC.

The main difference between using a POLFC and applying directly the planning result, is the availability, at time t, of a vector I_t of further information about the system. The optimization problem over the finite horizon h becomes:

$$
a_t^* = \arg \min_{a_t, p[t+1,t+h]} E_{\{w_\tau\}} \left[\sum_{\tau=t}^{t+h-1} g_\tau(x_\tau, a_\tau, w_\tau) + g_{t+h}(x_{t+h}) \right] \tag{3.16}
$$

That means to determine the optimal policy on the period $[t, t+h]$ using as a penal the optimal cost-to-go of the final state given by the solution of the planning problem; and using the noise distributions conditioned by the available information $I_t : w_t^{Da} \sim \phi_t^{Da}(\cdot|I_t), w_t^{Thao} \sim \phi_t^{Thao}(\cdot|I_t), w_t^{Lo} \sim \phi_t^{Lo}(\cdot|I_t)$.

Two remarks:

1. The length of the horizon h is decided mainly with regards to noise distribution: if $\forall \tau > \bar{h},\ \phi_t^{\cdot}(\cdot|I_t) = \phi^{\cdot}(\cdot)$, there is no point in the use of any $h \geq \bar{h}$, because the additional information I_t does not change the *a priori* knowledge of the system used in planning. Beside, as on-line policy must be calculated in a really short time, h must be reasonably small.

2. Solving problem 3.16, it is possible to consider directly one aggregated objective. This because the planning phase is already accomplished and the decision maker preference is already known through a λ weights-vector.

3.3 From the model to the MDP

Once defined completely the model, expressing it as a finite Markov Decision Process consists of two phases:

Discretization. To obtain a finite process, the state, control and inflow spaces must be discretized and finite. In this phase a classification for these spaces is defined.

Simulation. For all initial states, given controls and stochastic inflows, the evolution of the system has to be simulated, to obtain the final state, its probability and the pertaining costs.

The latter phase could be done through a Montecarlo simulation, which does not require a discretization of the inflows. Although, given the model of the inflows described in section 3.2.3, which admits that the stochastic distribution is known, the exhaustive simulation on the discretized inflows-space is more convenient.

3.3.1 Discretization

As described in section 3.2.4, the control description used for the system is already defined on a finite set $A = \{1, 2, \ldots, 27\}$. Differently, the system's state $x = (b, s, h)$ is defined in a 3-dimensional space $X = X_1 \times X_2 \times X_3$. Only the component X_1, which represent the bottom gates configuration b, is already discretized.

To decide the X_2 discretization, some alternatives have been tested through simulations. A set of 6 alternatives has been considered, ranging from 11 to 81 storage classes. Simulations have been run constraining the state s into the defined classes. The whole data set of 20 rain-seasons is used as inflows scenario, and three different policies are applied. The variation from the non-classified simulation, through the distances in cost values is evaluated. In figure 3.8 is showed the maximal cost distance from the non-classified simulation, over all the time-series and scenarios, for the different classifications. The 21-states discretization has been implemented.

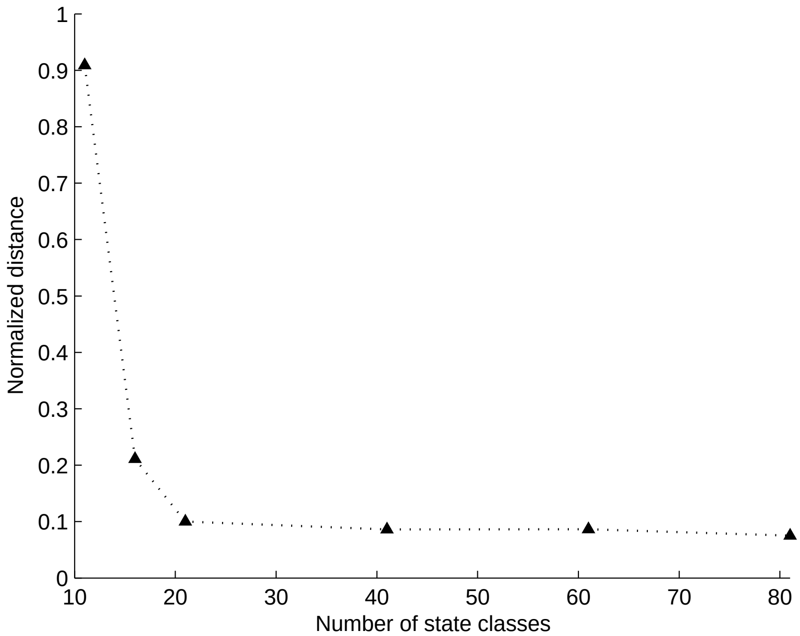


Figure 3.8: Maximal cost-distances from the continuous-state simulation

A similar approach to classify the X_3 space has been tried. However, the neural network used for the downstream model (see section 3.2.5), gives bad results with classified inputs. A different approach has consequently been used. Several classes-sets has been defined, and a new ANN trained for each one. Then the selection is done among these couple ANN-classes following three criteria:

Precision. The searched ANN must offer a good description of the system. It means that the estimations of the h-class have to be accurate.

Pessimism. The main purpose of the reservoir is to prevent flooding in Hanoi, and the ANN is describing exactly this part of the system. So a model that, also when not accurate, never underestimates the Hanoi level h is needed.

Smallness. To keep the state as smaller as possible, the previous objectives must be satisfied with the minor number of classes.

The result of this selection is a 9-classes discretization. In figure 3.9 is shown the validation result of the training. It is possible to see the good R result, and that the estimation errors committed are mainly overestimations.

The classes are described on table 3.3. It is noticeable that classes are more narrow around the flooding limit (+11.5m). This pattern showed consistently better results compared to equally distributed classes.

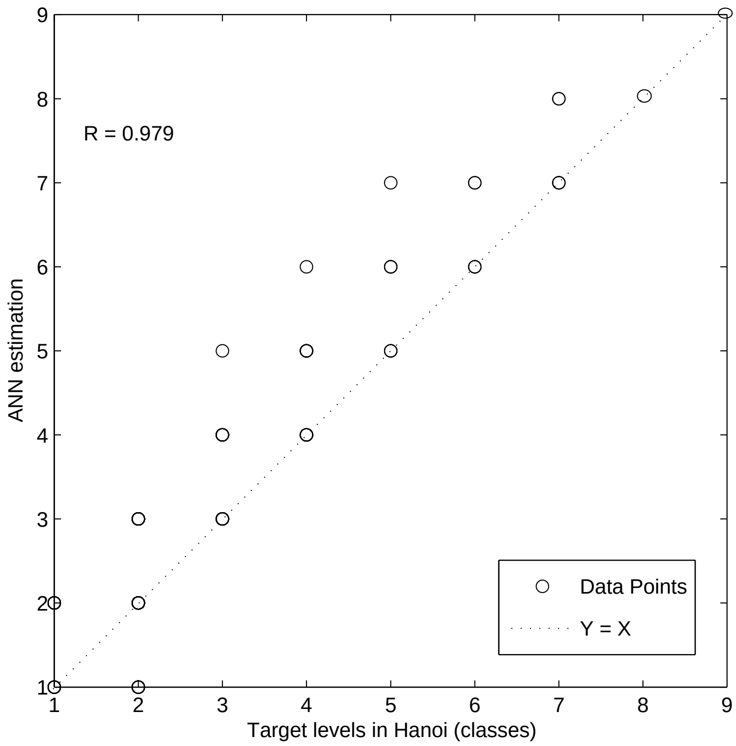


Figure 3.9: Validation results for the selected discrete ANN

Also the inflows-space is a 3-dimensional space $W = W^{Da} \times W^{Thao} \times W^{Lo}$. For each of the three dimensions a equally probable class definition is used. It is considered then, for simulation purposes, the .5-quantile of each class. The resulting W space is a set of equally probable inflows combinations $w = (w^{Da}, w^{Thao}, w^{Lo})$.

3.3.2 Simulation

The aim of the simulation is to determine:

$$p(y|x, a) \text{ is the probability transition: } X \times A \rightarrow X, \ \forall x, y \in X, \ \forall a \in A, \tag{3.17}$$

$$g(x, a) = (g^{DFlood}, g^{HyPow}, g^{UFlood}), \ \forall x \in X, \ \forall a \in A. \tag{3.18}$$

class	lower limit (m)	upper limit (m)
1	-	8
2	8	9.75
3	9.75	10.75
4	10.75	11.25
5	11.25	11.5
6	11.5	11.75
7	11.75	12.25
8	12.25	13.25
9	13.25	-

Table 3.3: Classes of set X_3

A series of one-stage simulation of the system, as described in section 3.2, are ran varying the initial state, the control and the inflows through all $X \times A \times W$, recording the costs generated and assigning the probability of $w \in W$ to the final state $y \in X$ of the single iteration.

The result of such operation are the functions $p(\cdot), g(\cdot)$ that, united to the sets $\{X, A, A(\cdot) \equiv A\}$, form the Markov Decision Process searched.

3.4 Validation

After the setting up of the MDP, a validation for some of the modeling choices is required. In particular the validation discussed here focuses on two aspects of the model:

1. The state discretization,
2. The control law structure.

More in detail, this validation aims to verify if a stationary policy, defined on the state classes listed in section 3.3, with decisions taken each 48 hours among the set A defined in 3.2.4, can obtain good performances when applied to the system. For this purpose, a comparison with the results obtained by Madsen et al. (2006) is described in this section.

In the cited article, a rule-curve optimization is performed, using GA. As stated in section 3.1.3, the actual operation rule and the optimized one are based on four key parameters (i.e.: water-levels in Hanoi and reservoir, inflows measures, season[9]), for which several thresholds are defined. A hierarchical check of parameters values (i.e.: a nested structure of "if. . . else. . . " statements) defines the decision to apply.

[9]The control law of the MDP discussed here is based only on the two water-levels. The POLFC could take into account the two other parameters.

In the optimization performed by Madsen and Le Ngo, seven of the thresholds are varied in a range, with the aim of minimizing two objectives representing "Hanoi flooding" and "hydropower potential deficit"[10], calculated for the four synthetic flood seasons described in section 3.2.3. Such time-series are showed in figure 3.10.

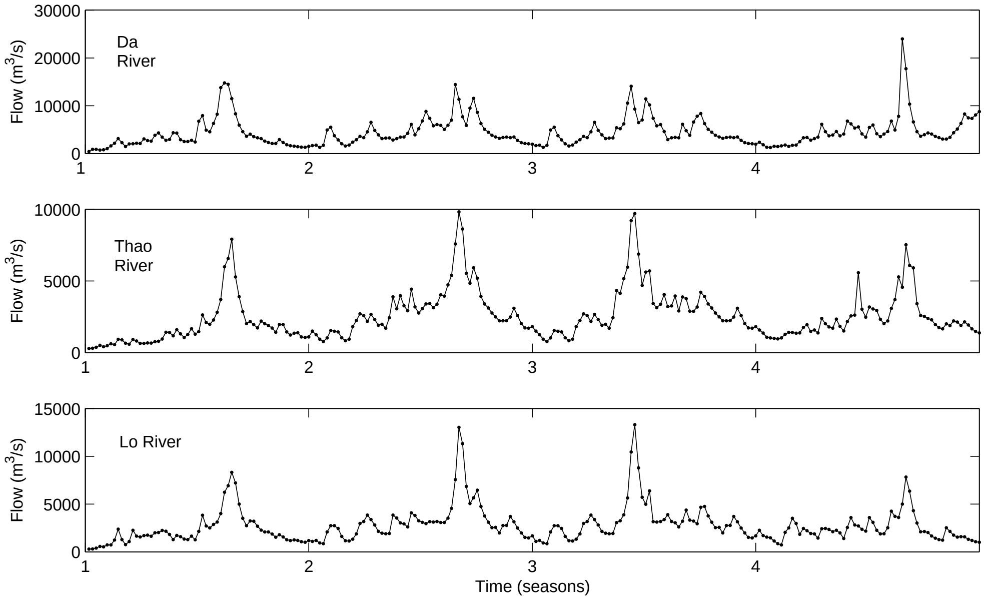


Figure 3.10: Synthetic series for validation (Madsen et al., 2006)

The two objectives minimized are[11]:

$$
\begin{aligned}
V^{DFlood,V} &= \sum_{i=1}^{4} \max_j (h_{i,j})^2 \\
V^{HyPow,V} &= \sum_{i=1}^{4} \frac{1}{T} \sum_{j=1}^{T} (117 - level(s_{i,j}))^2
\end{aligned} \quad (3.19)
$$

where $h_{i,j}$ is the level in Hanoi at the j-th stage of the i-th season, and $level(s_{i,j})$ is the water-level in the reservoir expressed in m, at the j-th stage of the i-th season.

The result achieved is a Pareto front for these two objectives, presented in figure 3.11. It can be noticed that the present regulation is largely dominated.

[10]Actually Madsen et al. (2006) performs a two-stage optimization: once obtained the results of the described minimization, another one is done varying other thresholds to improve two more objectives, related to hydropower production. Also the time-series used varies from one optimization to the other. In this thesis only the comparison with the first optimization is done. A second validation, requiring the creation of another MDP, should not bring further important information.

[11]The validation objectives are marked with a V to distinguish them from the optimization ones.

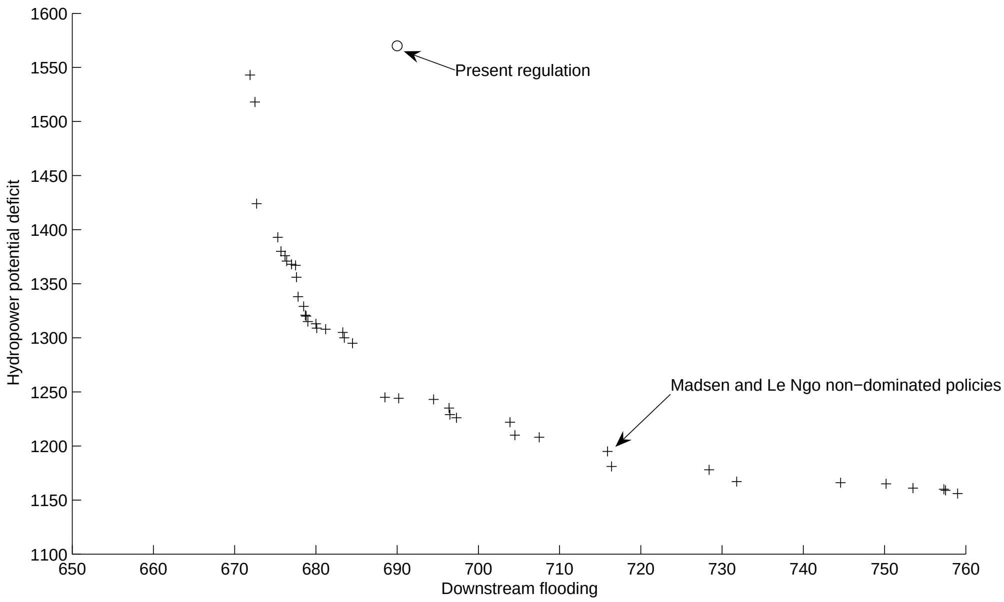


Figure 3.11: Pareto front proposed in Madsen et al. (2006)

The validation consists in an optimization of the model described in section 3.2, over the four synthetic seasons, minimizing the two objectives $V^{DFlood,V}$, $V^{HyPow,V}$.

One additional constraint has been added to the validation model: in present regulation, the control rule imposes to release at least the inflow quantity of water, when the level in the reservoir attains +117 m. This is made with the purpose of preventing a dam overflow. Madsen and his colleagues does not change this constraint in their optimization. To keep this constraint effective in the present validation, and because an exact application of it is impossible[12], it has been strengthen in the model imposing that for each reservoir storage excessing the +117 m level, all the release way must be opened. I.e.:

$$\forall x = (b, s, h) \in X : \ level(x) \geq +117m, \ A(x) = \{27\} \tag{3.20}$$

In the optimization model, as described in section 3.2.6, one further objective is defined to avoid reservoir's overflows. For such reason this constraint is not necessary on the optimization phase.

The validation model, with the constraint 3.20, has been optimized over the synthetic series. As this is not an MDP optimization, but a simulation-optimization approach, the Neighbour Search is

[12]The control law does not depend on inflows measures.

not applicable to this case. The method employed is the Noisy Cross-Entropy algorithm (Rubinstein, 1999; Dorini et al., 2006b). The application of the algorithm to this case will be the subject of a further paper. Here only the results are shown, in figure 3.12. The validation non-dominated policies corresponds to different objective weights; the policies signed as “Minimum downstream flooding”, “Compromise alternative”, and “Minimum hydropower deficit” are shown, as examples, in appendix A.

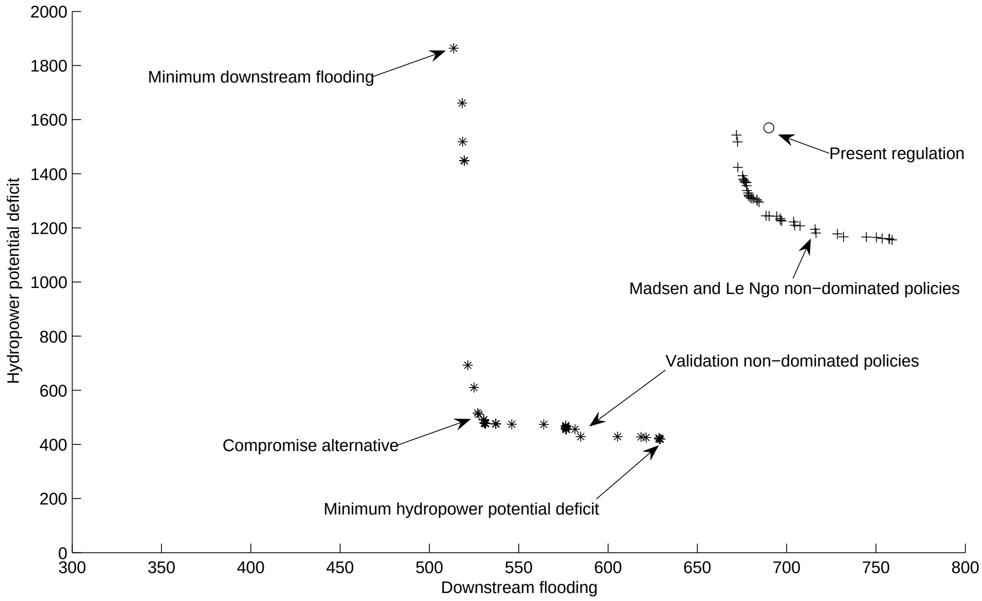


Figure 3.12: Results of the validation

Figure 3.12 shows that validation policies are, at least, never dominated by solutions by Madsen et al. (2006). Moreover some of the validation policies dominate, partially or entirely, the whole Pareto-front described in the cited article.

This result confirms that the proposed control law structure can perform better than the present regulation structure, also if the latter is improved through optimization. Also the discretization made of the state space is validated, as it gives a support to policies detailed enough to perform as shown.

3.5 Optimization

The achievement of the preceding phases is a Markov Decision Process with 1323 states and 27 available controls, describing the Hoa Binh system.

A DP optimization for a given weights-vector w requires an average of 6 seconds on the available computers. To calculate the Q-function for a policy φ through the policy-evaluation (PE) process (i.e.: iterating the equation

$$Q(x, u) = g(x, u) + \langle p(y|x, u), Q(y, \varphi(y))\rangle \tag{3.21}$$

until convergence) requires an average time of 4 seconds, including the Neighbor Search. In fact this process is computationally similar to DP, but generally needs less iterations to converge, as the policy is fixed.

However, in a 3 or more objective case, if two neighbours φ_1, φ_2 of a given point φ_3 are known with their Q-functions Q_1 and Q_2, the system

$$\begin{cases} \langle Q_1, w_A\rangle = \langle Q_3, w_A\rangle \\ \langle Q_2, w_B\rangle = \langle Q_3, w_B\rangle \end{cases} \tag{3.22}$$

can be considered, where w_A and w_B are the weights corresponding to the edges between φ_1 and φ_3 and between φ_2 and φ_3. If system 3.22 admits one and only one solution (i.e. iff the edges corresponding to w_A, w_B are not aligned), the Q_3 function can be computed solving it. Several efficient algorithms are available to find the solution of a linear system and such calculation requires, in the case considered, a time of magnitude 10^{-1} seconds allowing, when it can be applied, a relevant time gain.

It must be noticed that the solution of the equation 3.21 (the more time-consuming operation of the whole process) suffers of the curse of dimensionality, i.e. increasing the number of state-variables leads to an exponential growth of the calculation-time. In the other hand increasing the number of time steps, using a cyclic system, causes only a linear increase of calculation-time, as for DP. However, applying the Neighbour Search in cyclic system cases requires to operate on the enlarged-state Q-function (i.e.: including the time variable). As several Q-function must be stored during the optimization, in order to exploit as often as possible the system 3.22, memory requirements consistently increase. If memory availability does not support such increase, the optimization process speed decrease at a rate larger than linear, although this is theoretically possible.

Chapter 4

Results

In this chapter, the optimization results (i.e.: the Pareto-front of the MDP)is presented in detail. Further some discussion on the advantages offered by the complete knowledge of the Pareto-front, enabled by the NS is given through a set of examples.

4.1 The Pareto-front

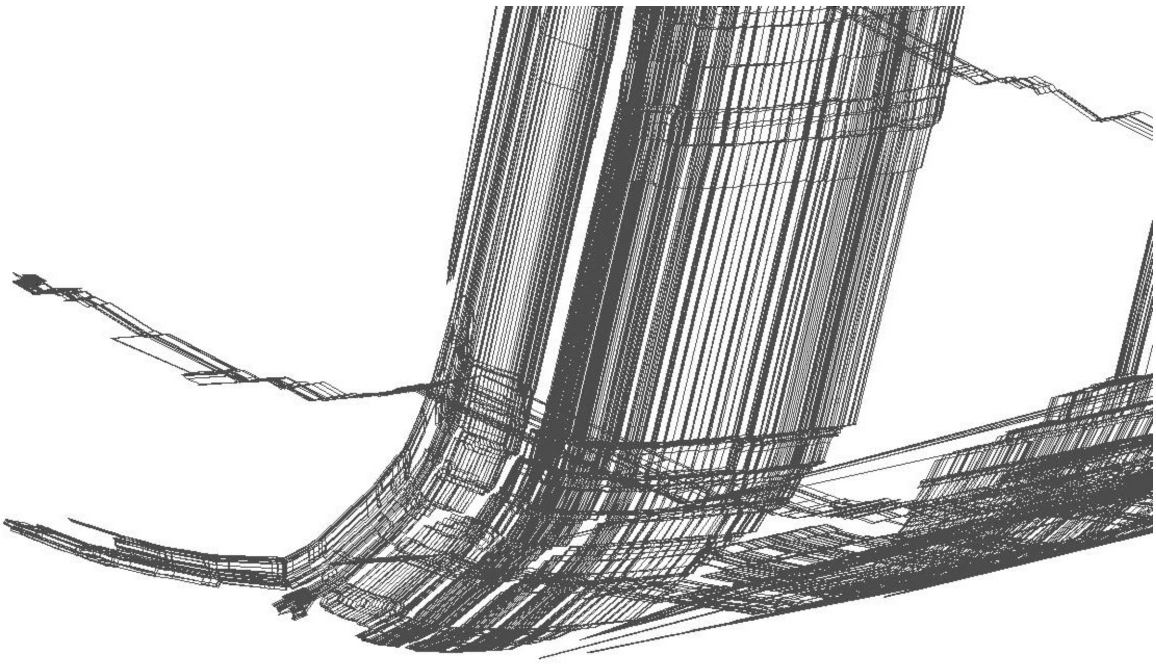


Figure 4.1: Detail of the Pareto front

In figure 4.1 is represented a part of the Pareto-front resulting from the optimization. It is to remark that the Pareto-front is formed by *faces*, as described in chapter 2. In what follows, however, for ease of readability , the front will be represented by points, corresponding to the stationary policies (i.e.: the vertices of the faces). In figures 4.2, 4.3, 4.4 the three bi-dimensional projections of the front are presented with this criterion.

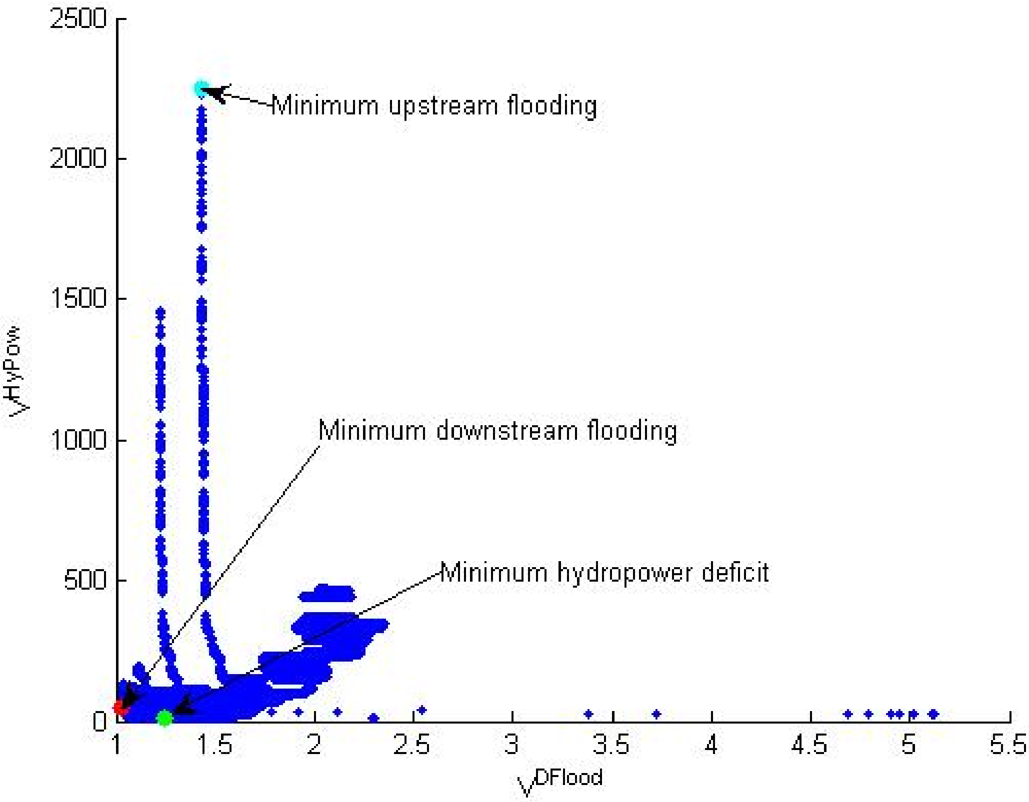


Figure 4.2: The Pareto front - Projection on V^{DFlood} and V^{HyPow}

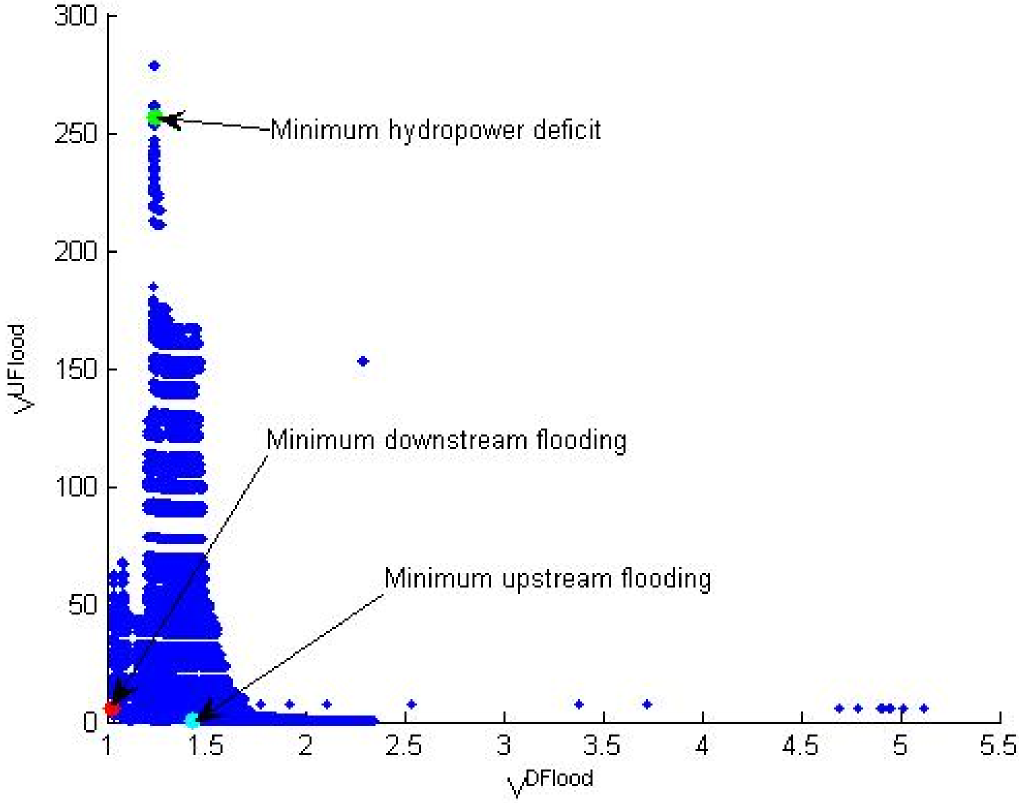


Figure 4.3: The Pareto front - Projection on V^{DFlood} and V^{UFlood}

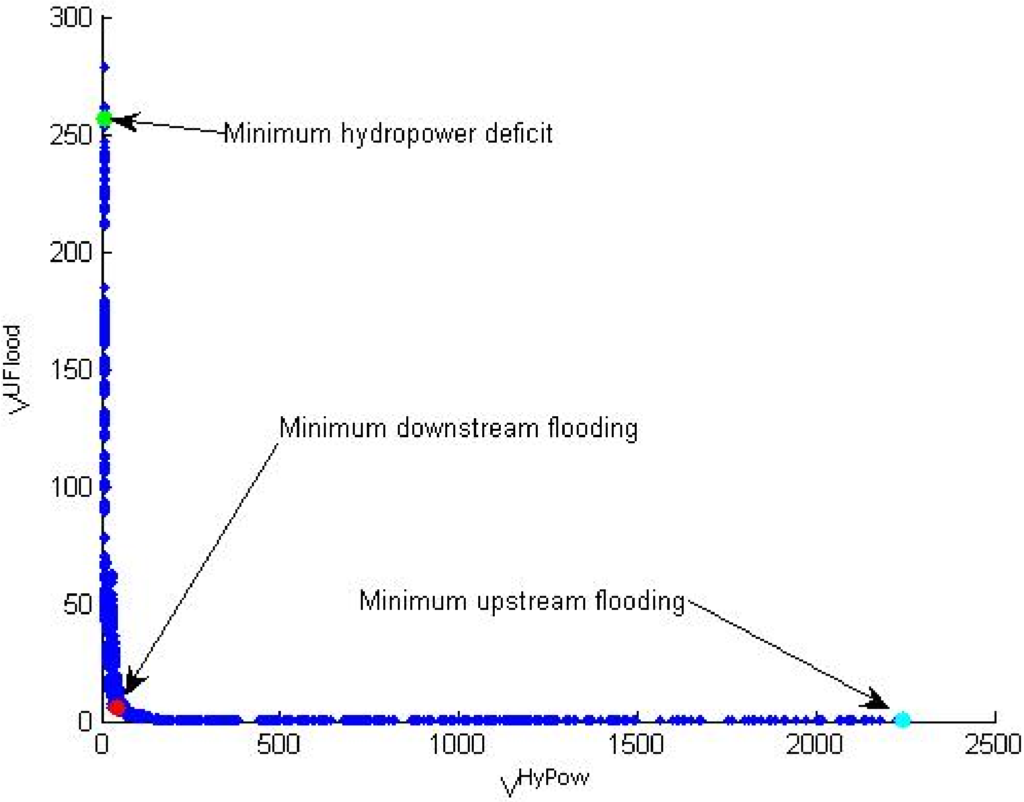


Figure 4.4: The Pareto front - Projection on V^{HyPow} and V^{UFlood}

The analysis of the solutions providing the optimum for each individual objective represents the first step toward the policies' performance assessment. This usually provides insight to understand patterns with regards to the different purposes of the policies. Such points are highlighted in the front projections, and the corresponding policies are detailed in appendix A (sections A.2.1, A.2.2, A.2.3). The policies are there shown both explicitly (as tables) and through their simulated application to the 20-years long available time series[1].

Some preliminary observation can be made, comparing the extreme policies for V^{HyPow} and V^{UFlood} with the V^{DFlood} one, :

- The policy minimizing V^{HyPow} tends to keep the reservoir quite full, releasing as often as possible only through the eight turbines: as it can be observed from simulated trajectories, in several years the water storage reach the maximum allowed. This behavior is dangerous both for the dam and for Hanoi. In fact, one more flooding is registered downstream (season 2) that could be avoided applying the V^{DFlood} optimal policy.

- The optimal policy for V^{UFlood} shows an opposite trend: it maintains the water level in the reservoir as low as possible. This also causes the Hanoi flooding in the second season, but only with the aim of keeping the storage capacity unused. In the remaining seasons, this policy performs well regarding to the downstream flooding control. It yields a consistent increase of hydropower deficit.

  It could be argued that the degradations shown by the first and second objectives when this policy is applied are not outweighed by the lowering of the reservoir level, that is kept always far from the attention threshold. The decision-maker could be probably interested in policies that exploits more the reservoir storage capacity, in order to protect Hanoi and produce electricity.

As it was expected, V^{HyPow} is conflicting with the two other objectives. This because it increases with the available flow through turbines and with the water level in the reservoir (essentially for its effect on headwater). The conflict between V^{HyPow} and V^{UFlood} is illustrated in figure 4.5 by means of a parallel coordinates plot (Wegman, 1990), where several policies are represented as lines, connecting the values of the objectives of the policy.

The intersections are a visual measure of the conflict: if, moving from one alternative to another one objective increases while the other decreases (i.e.: the objectives are conflicting), the lines corresponding to the alternatives have opposite orientations. The parallel coordinates plot of non-conflicting objectives would show non-intersecting lines.

[1]This is just one of the possible ways to illustrate a policy, as it is possible to calculate different performance parameters that synthesize the simulation (i.e.: maximal water levels, number of days of flooding in a period,...) or it is also possible to run Markov simulations. This offers an evaluation of how the policy affects the MDP. Also selected scenarios could be effective for a policy representation. The choice to use the historical time-series simulation is made in order to give to the decision-maker a simple and readable information, considering that nothing is known about his/her preferences.

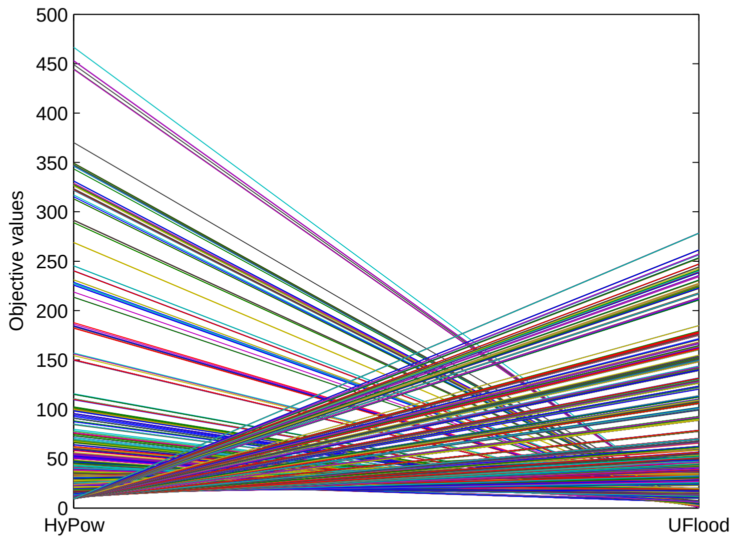


Figure 4.5: Parallel coordinates plot for the objectives V^{HyPow} and V^{UFlood}

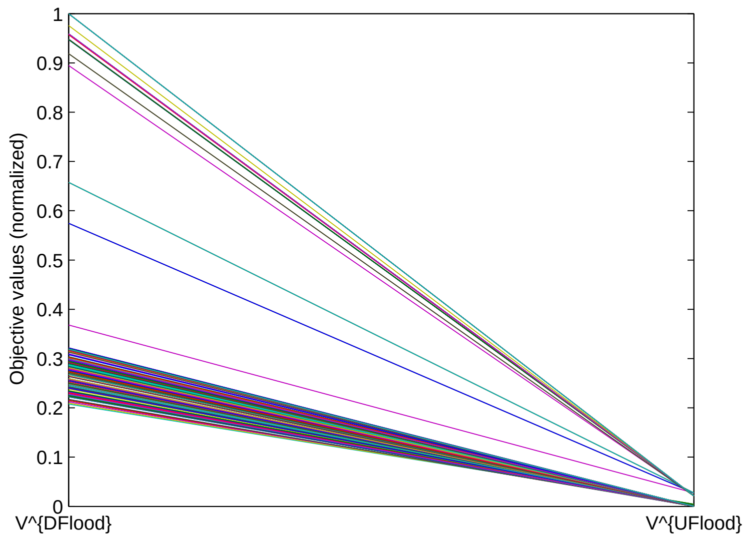
More interesting is the conflict between V^{UFlood} and V^{DFlood}, which is illustrated in figure 4.6. For high values of the objectives (in the plots they are normalized in a 0 to 1 range) they are non-conflicting (i.e.: keeping the reservoir level low helps to prevent Hanoi flooding, and viceversa). However, for solutions with good performances, the two objectives conflict.

Such behavior can be explained through the following observation: until the water-levels (both in Hanoi and in the reservoir) are non-critical, any optimal policy will release enough to keep the reservoir storage low, but not so much to produce a downstream flooding. However, when a critical situation occurs, the differences between policies will become significant: a policy rewarding V^{DFlood} will store into the reservoir as much water as possible to try avoid Hanoi flooding, even if it will result in a critical level for the dam. On the contrary, a policy with a lower V^{UFlood} will release any quantity of water necessary to keep the reservoir safe, also causing downstream floods.

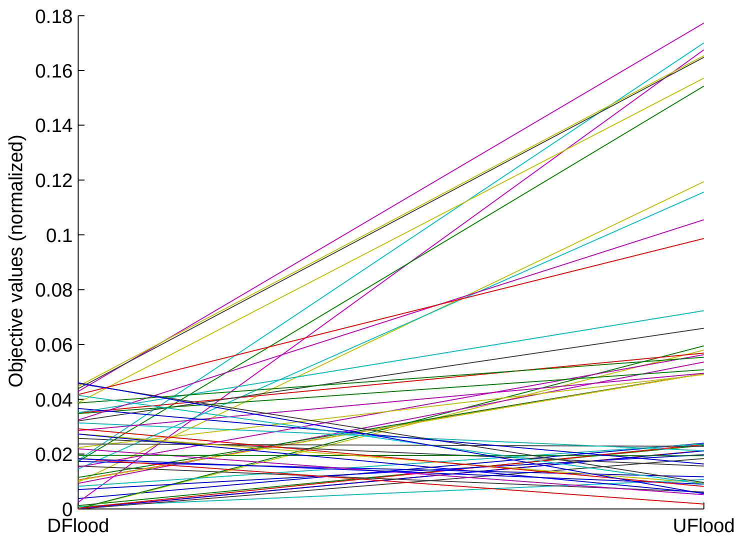
4.2 Exploration of the Pareto-front

Building on the considerations presented in the previous section, it is possible to explore the Pareto-set searching for one or more potential compromise solutions. Since the decision-maker (i.e.: the vietnamese government) was not directly involved, this exploration is given just as a reasonable example of a decision process based on the NS results.

Let us consider figure 4.4. This projection of the front shows two large and extremely steep



(a) High values of V^{UFlood}



(b) Low values of V^{UFlood}

Figure 4.6: Parallel coordinates plots for the objectives V^{Dflood} and V^{UFlood}

extremes, and a relatively small "knee area". The steepness of the front can be read in terms of trade-off between objectives: in a nearly horizontal or vertical part of the Pareto-front, one objective can be consistently improved with a small degradation of the other. In general, it is reasonable to assume that the decision maker is more interested in the portions of the objective space with low marginal rates of substitution; moreover, for the case study considered, the extreme point for the objective non-represented (i.e.: V^{DFlood}) is situated in that lower corner area. This offers an additional reason to focus on this part of the front.

However, it is necessary to verify that the consistent increase of V^{UFlood} does not cause an unacceptable risk for the reservoir. To evaluate this risk and the variations in electrical production, two points have been chosen on the edges of the corner area, and the corresponding policies were simulated. The points are marked P1 and P2 in figure 4.7; the simulated reservoir levels and hydropower deficits are plotted in figures 4.8 and 4.9.

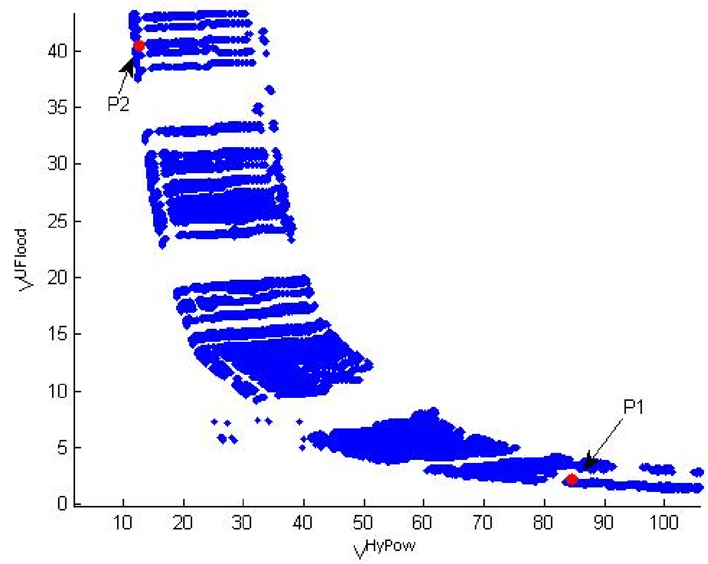


Figure 4.7: Detail of the Pareto-front projection on V^{HyPow} and V^{UFlood}

Quite expectedly, both P1 and P2 show an improvement in power generation, compared to the minimal V^{UFlood}. However, the difference between the two points is not so consistent, and it is concentrated in heavy rain seasons (e.g.: season 1). At the same time, if P1 keep the water-level far enough from the upper limit, moving from P1 to P2 causes the storage to reach this limit two additional times, on the historical scenario. The advantage in power generation does not seem to justify the increased risk. P1 is then assumed as a good compromise alternative between V^{HyPow} and V^{UFlood}.

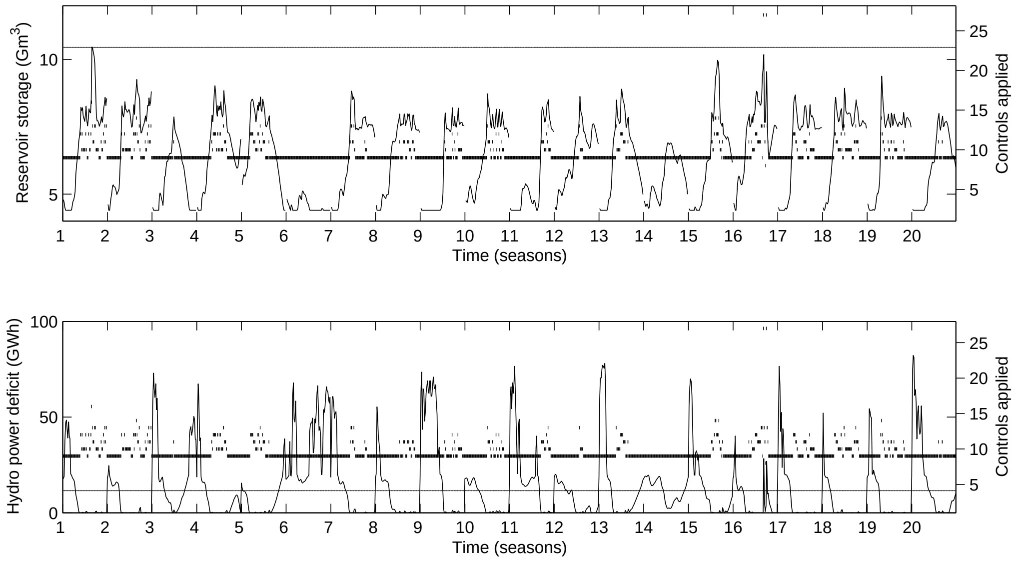


Figure 4.8: Simulation of P1

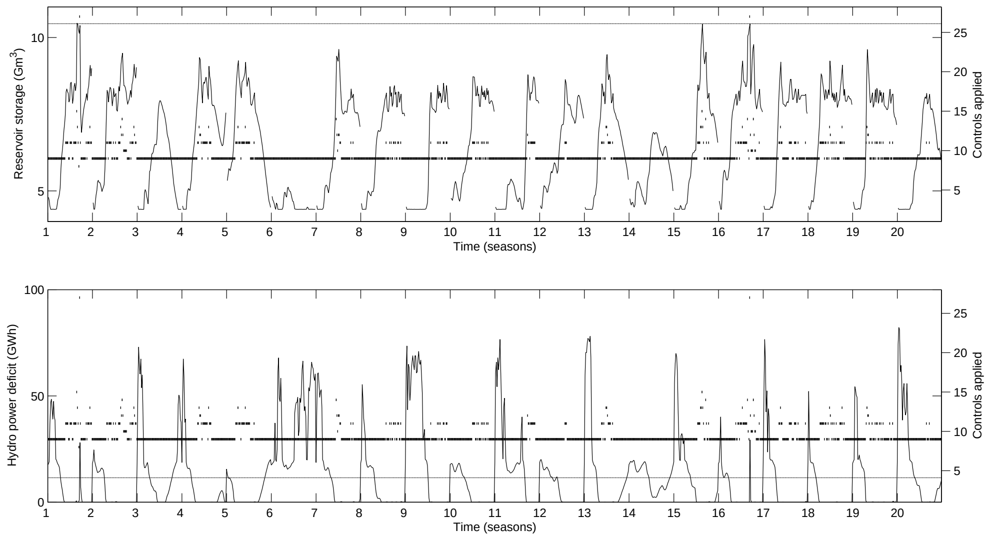


Figure 4.9: Simulation of P2

The decision maker may not be satisfied by the downstream flooding performance of the selected policy P1, and he may desire to reduce the performances on the two other objectives to improve that one. In this circumstance the *a priori* knowledge of the Pareto front in its entirety offers “” advantage.

Figure 4.10 shows projections of the Pareto front on V^{DFlood} and V^{HyPow} corresponding to different levels of the V^{UFlood} objective, ranging from 2 to 4 with a .2 interval. The policy P1 is also highlighted.

Three new policies are explored from P1[2]:

- P3 is on the iso-V^{UFlood} curve passing through P1. It corresponds to an improvement in V^{DFlood} obtained without losses on the dam-protection objective.

- P4 has the same V^{DFlood} value than P3. The same improvement in the objective obtained through P1→P3 is obtained through P1→P4, but with losses only in V^{DFlood} and not in hydropower production.

- P5 is obtained choosing an intermediate point between P3 and P4. It has the same V^{DFlood}, but losses are distributed between V^{HyPow} and V^{UFlood}.

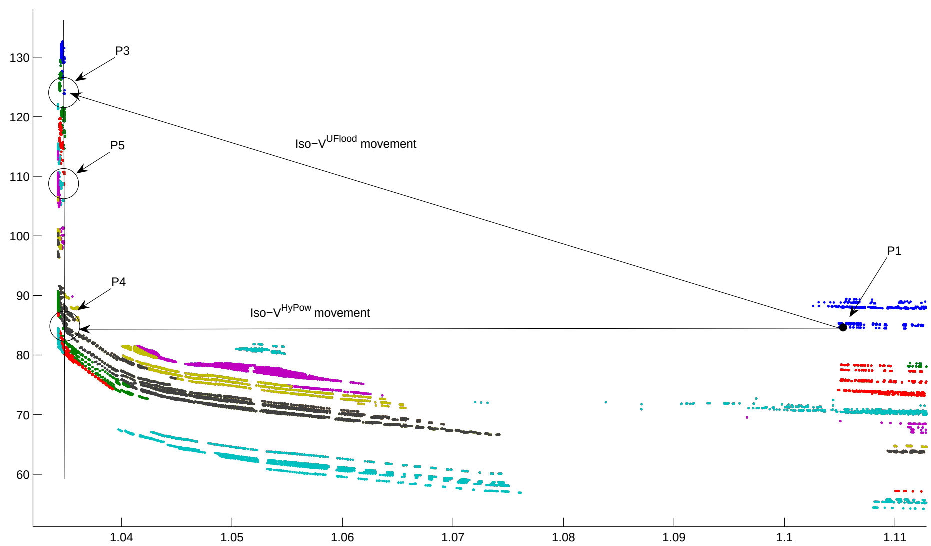


Figure 4.10: Pareto-front projection on V^{Uflood} and V^{HyPow}.

[2]The coordinates of the points are reported in table 4.1, and the corresponding simulations are shown in appendix A.

Point	V^{Dflood}	V^{HyPow}	V^{Uflood}
P1	1.1053	84.632	2.1842
P3	1.0348	124.436	2.1535
P4	1.0348	84.633	3.4862
P5	1.0348	108.792	2.5667

Table 4.1: Coordinates of points P1, P3, P4, P5

Only minor differences can be noticed among the three policies' performances on the historical scenario simulations. For example, applying P3 or P4 causes a peak in power generation deficit in season 15, which is avoided applying P5.

In order to appropriately compare these alternatives, a larger set of evaluation criteria would be required, involving synthetic series, Markov simulations, or other more sensitive parameters. However, such fine comparison is not within the scope of this thesis.

Another advantage offered by NS approach, useful for decision "fine tuning", is the knowledge of the neighbourhood of all policies: given a selected point of the Pareto-front, all its neighbours can be extracted and evaluated. Moreover, any policy with a performance lying in the neighbourhood can be generated through randomization. An example of neighbourhood is plotted for the policy P5 in figure 4.11

4.3 Discussion

The Neighbor Search technique was shown to enable the exploration of the Pareto front efficiently and exhaustively. In the case study considered, such method founded more than 1 million optimal policies and all their geometrical relationships in a reasonable time.

The concept behind NS is to generate simultaneously, at each iteration of the algorithm, both a point of the Pareto-front and the directions to find the next ones. This is possible, for a defined class of problems including finite MDP, exploiting geometrical properties of the Pareto-front itself.

Traditional single-objective optimization techniques provide no information about exploring directions in the Pareto-set. The limits of such approaches emerge when the dimensions of the searched front increase, in terms of number of points and of objectives.

Multi-objective methods, such as MOEAs, can find several solutions per run. However, it is not straight-forward to obtain from such heuristic methods accurate information about the Pareto-front: nothing is known about the front's shape between two points, and a non-trivial tuning is necessary in order to obtain a good sampling of the Pareto-set. Moreover, only in particular cases it is possible to guarantee that the solutions found really belong to the Pareto-set and are not only sub-optimal.

The knowledge of the entire Pareto-front and of its geometrical structure (vertices, edges and faces) substantially improves the decision-making: given two or more policies, it is always possible to know if other policies exists with performances lying on the hyperplane they define, and how many they are. Moreover, if no such policies exist, the edge or face supporting them is known, and it is possible to achieve any performance along this segment using randomized policies. This clearly facilitates the negotiation process between conflicting stakeholders.

NS offers also a further advantage compared to other techniques: as described in chapter 2, it generates for each policy found the directions of all the neighbours, which in further iterations are explored following a random order. It is possible to intervene on such order to specify a directional priority for the exploration process.

It is therefore possible to explicit such information within an interactive search, whereby decision maker and stakeholders preferences are elicited iteratively to direct the exploration and focus on subportions of the Pareto-front.

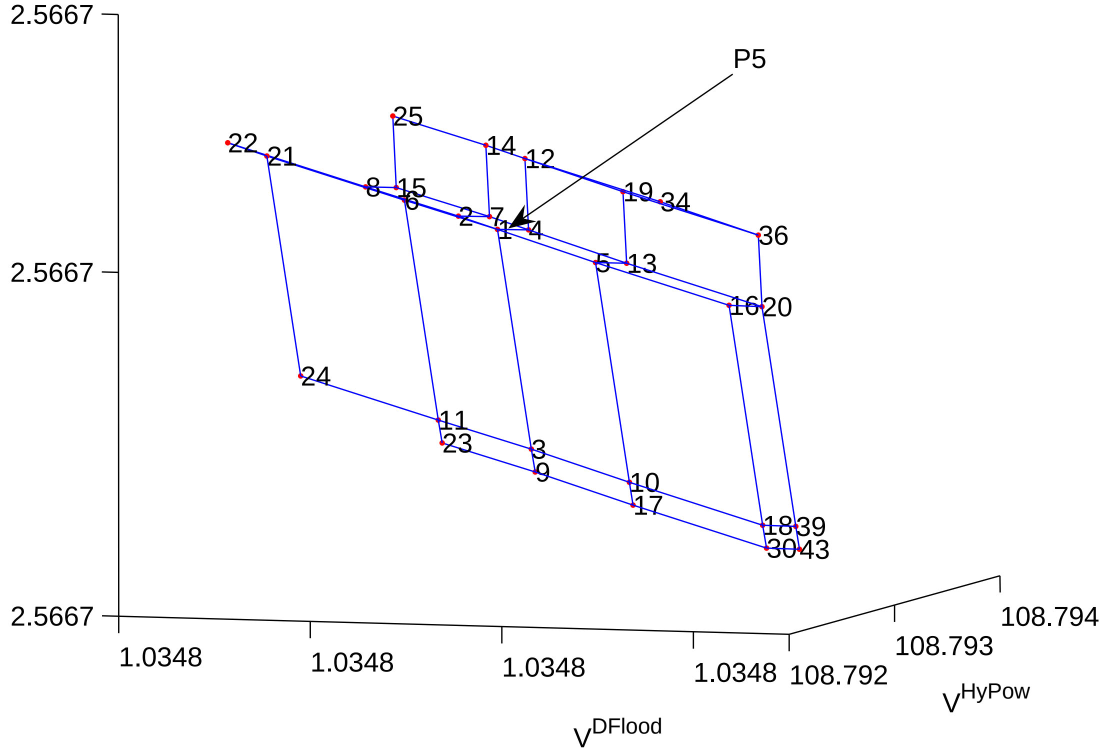


Figure 4.11: The Neighborhood of the policy P5

Chapter 5

Conclusions

The focus of this thesis is to show, through a real-case application, how the NS can be useful to improve the decision-making process for multi-objectives reservoir operation planning.

After a survey of the principal techniques employed in literature to solve such problems, the NS algorithm has been discussed.

Further, the case-study has been presented. Hoa Binh is the largest reservoir in Vietnam, providing 40% of its total power supplies and protecting the capital Hanoi from major flooding events. This double purpose generates conflicts in its management planning.

A model of the system has then be detailed. A control law giving 48-hours release-ways regulation on the basis of water-levels in the reservoir and in Hanoi, and of the dam's bottom-gates configuration has been implemented. Inflows have been represented as a pure stochastic process while downstream model is simulated through an auto-regressive artificial neural network.

Further, the model state-space and noise-space have been discretized in order to describe the system as a finite Markov Decision Process.

The modeling phase is concluded by a validation of the control law structure and of discretization. Such validation required to optimize the implemented control law over a deterministic severe flooding scenario provided by Madsen et al. (2006). The results obtained in this phase, compared with those obtained by the cited authors, confirmed the modeling approach used.

Once the model of the system validated, the corresponding MDP has been optimized applying the NS algorithm. This optimization provided more than 1 million policies of the Pareto-front, and the geometrical relationships among them.

Building on these results, an example of Pareto-front exploration was illustrated. The aim of such exploration was to provide instances of the analysis allowed to extract selected policies from the front:

- extraction of points through a whole-front search based on objective values (i.e.: the single-objective extremes),
- extraction of points based on the Pareto-front positioning through a visual selection (i.e.: P1 and P2, on the edges of the $V^{HyPow} - V^{UFlood}$ low marginal substitution rate area),
- extraction of points through constrained movement on the objective space, both with mono and bi-dimensional constraints (i.e.: P3 and P4 constrained to a 1D space, P5 to a 2D).
- extraction of the neighbors of a selected point using faces informations.

The first three category of extractions are possible because all the Pareto-front is known. Such explorations are then theoretically feasible, but not straight-forward, also using traditional optimization techniques. On the contrary, the last one is obtained through the geometrical properties of the front, which are not provided by any other technique.

Some of the extracted policies have been simulated on the historical scenario, to asses their performances.

Finally, it has been highlighted how the NS approach generates and uses more information than traditional techniques at each iteration of the optimization process, and how this feature represents an advantage for the decision-making process.

The case-study presented in this thesis represents a first step toward the development of a competent DSS based on the NS, and it suggests some future research areas.

The main area to investigate is the possibility of integrating the NS into an interactive optimization method, thereby enabling the selected exploration of the Pareto-set under the control of the decision-maker. This method could be useful in applications that involve large state systems, for instance multi-reservoir optimization, allowing a fast exploration of most interesting areas. In fact, although a rule to *a priori* determine the number of efficient policies for a model has not be found yet, when the dimension of the system increases, this number normally also increases, and so does the number of iterations required by NS. In such cases a method to focus on sub-portions of the Pareto-front would be desirable.

The application of the Neighbor Search approach to the operation optimization of the Hoa Binh reservoir has been used as a benchmark for this new algorithm. It demonstrates that such algorithm is an efficient and exhaustive way to find the optimal solutions for this class of problems and that it substantially improves the decision-making process. One of the most important feature that make NS approach so useful for decision processes is the additional geometric information this technique provides about the Pareto-front.

In conclusion, the results presented motivate further research into the application of the NS to water management problems. In fact, in the authors' opinion, the NS could be successfully used to achieve a more efficient, conscious and participated use of such an important natural resource.

Appendix A

Policies

In this appendix the policies described in the text are shown through simulation and tables. The simulations of *validation* policies (see section 3.4) are calculated over the four synthetic season described by Madsen et al. (2006). The Pareto-optimal policies for the MDP (see chapter 4) are simulated over the whole 20-seasons time-series. For each simulation the controls applied are shown, together with the reservoir's inflows and storage, and the level in Hanoi. For optimal policies, also the hydropower deficit for each time-step is plotted.

How to read the policy tables is explained in figure A.1.

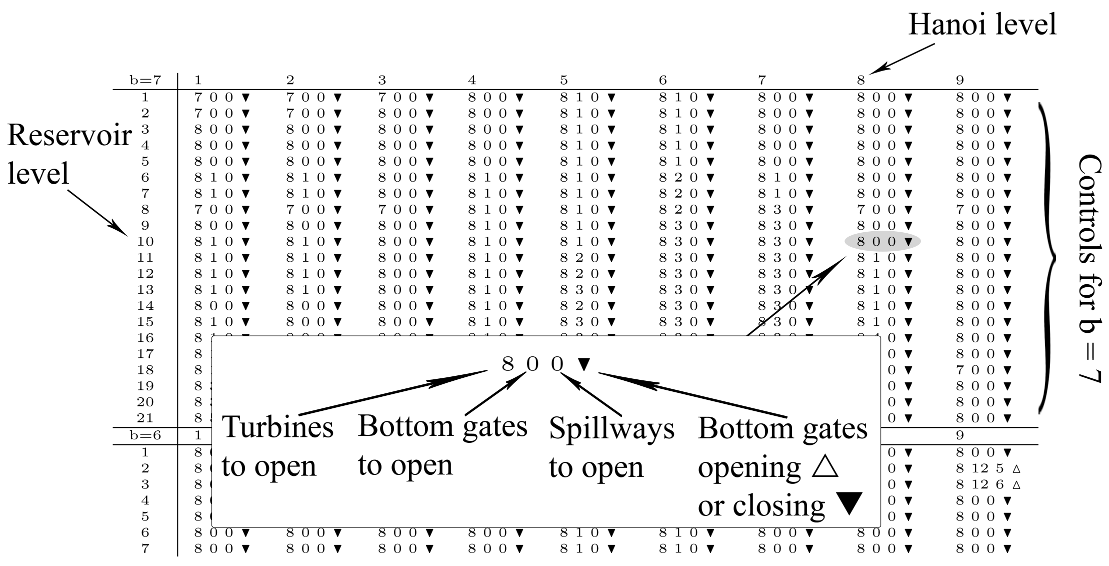


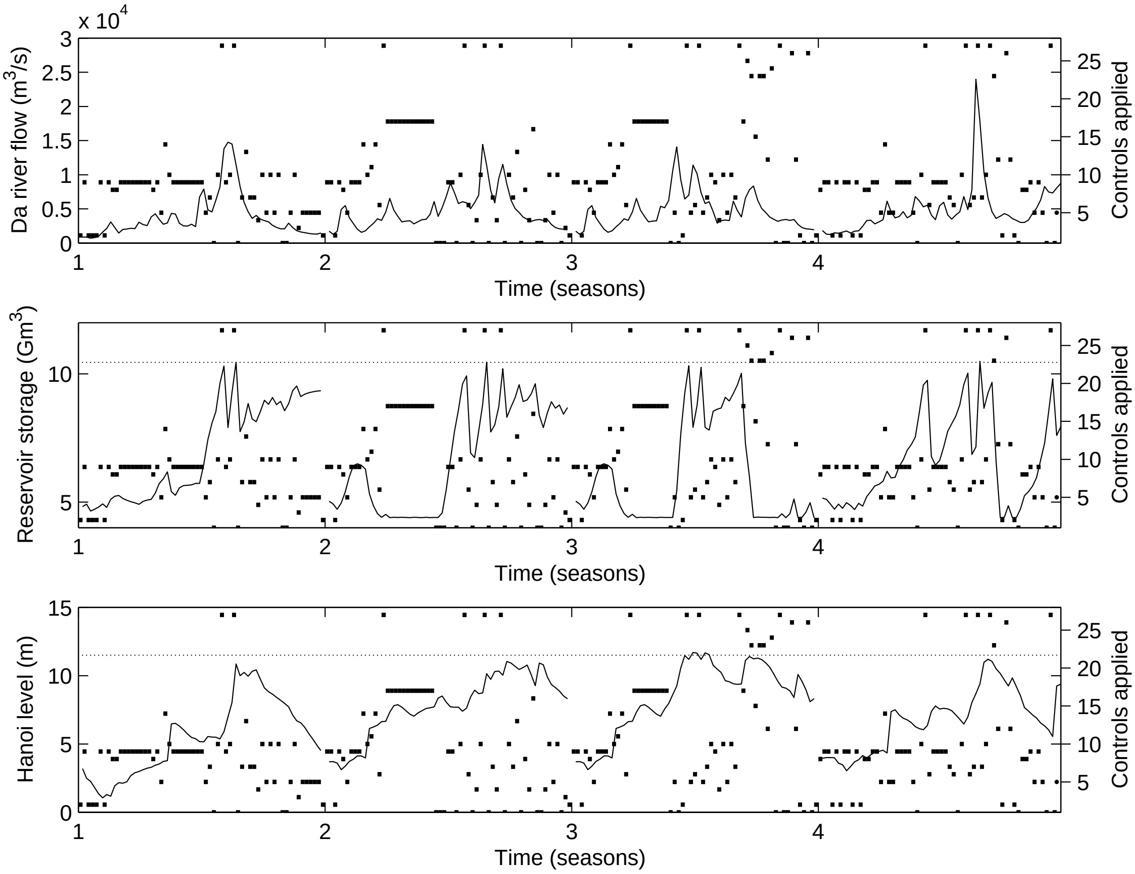
Figure A.1: How to read policies

A.1 Validation policies

The three policies shown in this section are results of the validation process presented in section 3.4.

It can be noticed from the tables how these policies do not show continuous trends varying the state. The reason of this behavior can be found in the optimization process that generated them. In fact, the validation policies are the result of a deterministic simulation-optimization approach on a given scenario (see figure 3.10). The control laws are consequently optimized only for the states reached through the simulation. To obtain policies optimized for all states it is necessary to use a different approach such as DP or, in this case, NS.

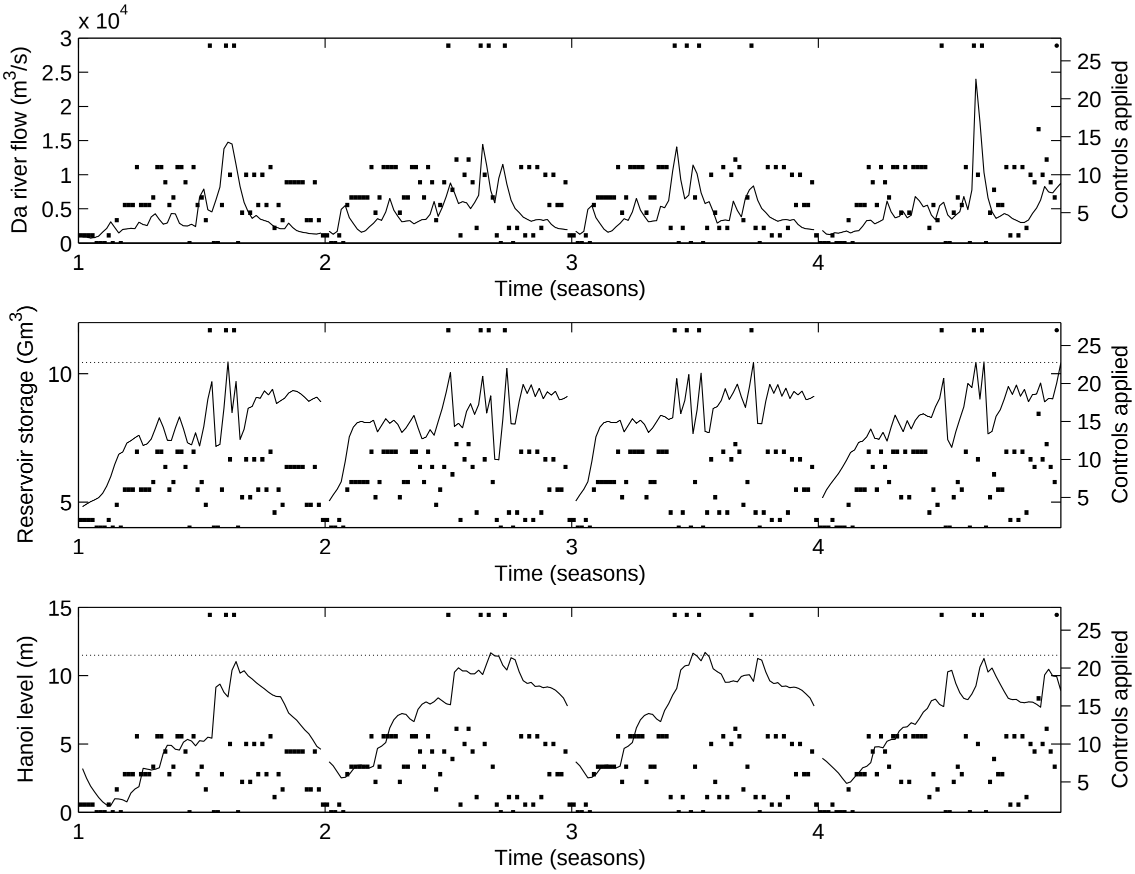
A.1.1 Minimum downstream flooding



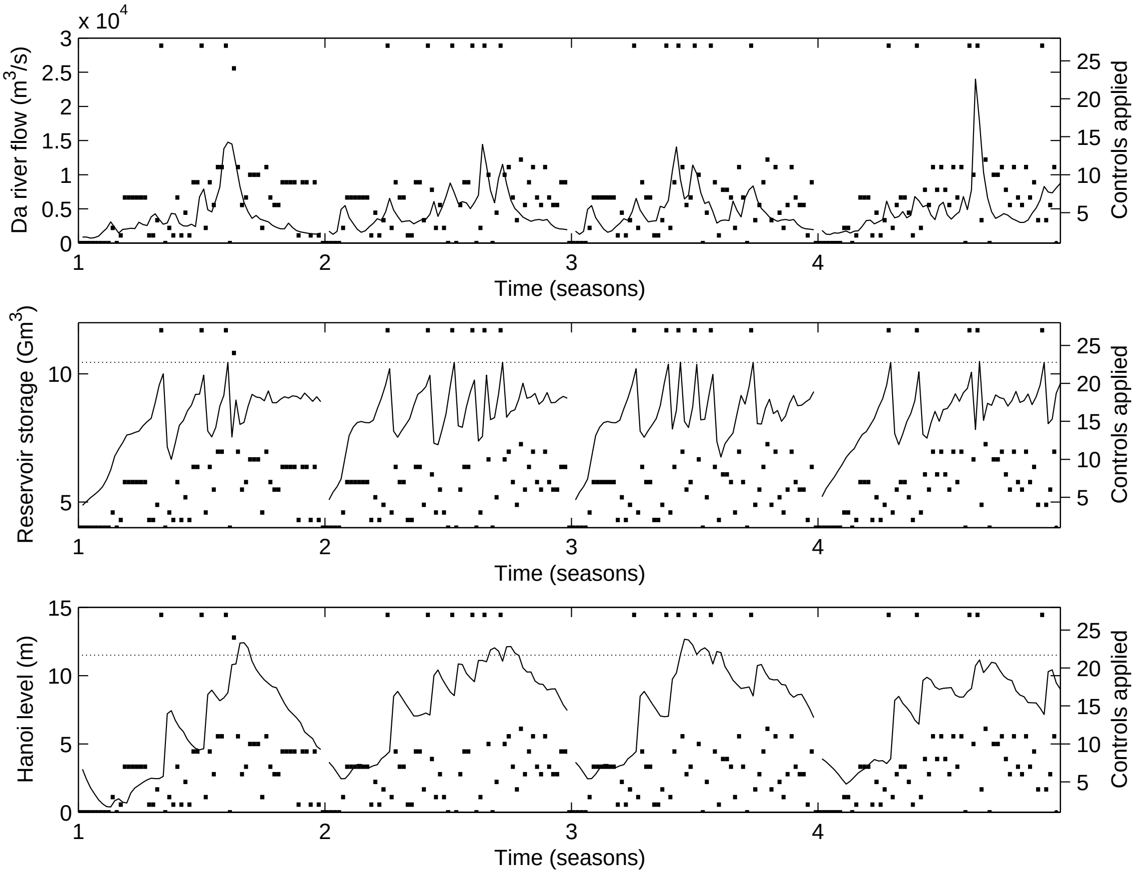
b=7	1	2	3	4	5	6	7	8	9
1	8 8 0 =	lock ▼	8 3 0 ▼	8 12 2 =	8 6 0 =	2 0 0 ▼	8 9 0 =	6 0 0 ▼	8 7 0 =
2	2 0 0 ▼	lock ▼	7 0 0 ▼	8 0 0 ▼	7 0 0 ▼	8 12 2 =	8 3 0 ▼	8 3 0 ▼	1 0 0 ▼
3	3 0 0 ▼	8 12 1 =	8 7 0 =	8 0 0 ▼	8 9 0 =	8 12 1 =	8 12 4 =	8 2 0 ▼	8 1 0 ▼
4	1 0 0 ▼	7 0 0 ▼	8 12 4 =	6 0 0 ▼	8 12 4 =	8 5 0 ▼	8 1 0 ▼	8 7 0 =	7 0 0 ▼
5	8 6 0 =	8 2 0 ▼	8 2 0 ▼	8 12 5 =	8 6 0 =	8 12 5 =	8 12 0 =	8 2 0 ▼	8 11 0 =
6	8 3 0 ▼	8 5 0 ▼	4 0 0 ▼	8 12 1 =	8 12 4 =	8 5 0 ▼	6 0 0 ▼	lock ▼	8 12 5 =
7	8 12 0 =	8 12 3 =	8 12 6 =	lock ▼	2 0 0 ▼	8 3 0 ▼	8 6 0 =	8 12 2 =	1 0 0 ▼
8	8 10 0 =	8 2 0 ▼	8 12 2 =	3 0 0 ▼	1 0 0 ▼	8 10 0 =	lock ▼	8 9 0 =	8 10 0 =
9	5 0 0 ▼	5 0 0 ▼	2 0 0 ▼	8 12 6 =	4 0 0 ▼	8 9 0 =	8 12 3 =	7 0 0 ▼	8 6 0 =
10	8 7 0 =	3 0 0 ▼	8 12 0 =	lock ▼	8 9 0 =	8 10 0 =	8 2 0 ▼	8 12 3 =	8 3 0 ▼
11	8 1 0 ▼	lock ▼	8 11 0 =	8 8 0 =	7 0 0 ▼	6 0 0 ▼	6 0 0 ▼	lock ▼	lock ▼
12	4 0 0 ▼	lock ▼	lock ▼	lock ▼	8 1 0 ▼	2 0 0 ▼	7 0 0 ▼	8 11 0 =	lock ▼
13	8 0 0 ▼	8 12 4 =	8 6 0 =	lock ▼	8 3 0 ▼	4 0 0 ▼	8 2 0 ▼	8 7 0 =	8 9 0 =
14	8 7 0 =	8 12 2 =	5 0 0 ▼	lock ▼	8 5 0 ▼	6 0 0 ▼	6 0 0 ▼	8 12 0 =	8 4 0 ▼
15	8 10 0 =	8 6 0 =	lock ▼	6 0 0 ▼	8 12 4 =	3 0 0 ▼	6 0 0 ▼	8 7 0 =	7 0 0 ▼
16	8 7 0 =	8 0 0 ▼	6 0 0 ▼	8 4 0 ▼	8 12 2 =	5 0 0 ▼	lock ▼	8 12 3 =	8 8 0 =
17	8 7 0 =	8 11 0 =	1 0 0 ▼	8 8 0 =	8 6 0 =	2 0 0 ▼	8 4 0 ▼	6 0 0 ▼	8 12 2 =
18	5 0 0 ▼	8 12 0 =	8 3 0 ▼	8 12 6 =	8 5 0 ▼	8 10 0 =	8 5 0 ▼	8 1 0 ▼	1 0 0 ▼
19	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =
20	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =
21	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =	8 12 6 =
b=6	1	2	3	4	5	6	7	8	9
1	5 0 0 ▼	8 12 2 △	2 0 0 ▼	8 11 0 △	8 3 0 ▼	7 0 0 ▼	7 0 0 ▼	8 7 0 △	8 12 4 △
2	8 9 0 △	8 8 0 △	8 12 4 △	8 6 0 △	2 0 0 ▼	8 9 0 △	8 12 1 △	8 5 0 =	8 9 0 △
3	8 1 0 ▼	4 0 0 ▼	8 8 0 △	8 9 0 △	8 7 0 △	3 0 0 ▼	8 10 0 △	8 12 4 △	1 0 0 ▼
4	8 1 0 ▼	lock ▼	8 9 0 △	8 12 2 △	8 12 5 △	8 5 0 =	8 6 0 △	8 1 0 ▼	8 6 0 △
5	8 2 0 ▼	8 2 0 ▼	8 12 6 △	8 5 0 =	4 0 0 ▼	8 6 0 △	5 0 0 ▼	8 5 0 =	8 11 0 △
6	4 0 0 ▼	8 6 0 △	8 1 0 ▼	7 0 0 ▼	8 6 0 △	8 12 0 △	8 1 0 ▼	8 11 0 △	8 8 0 △
7	8 4 0 ▼	8 12 2 △	6 0 0 ▼	8 12 3 △	8 1 0 ▼	1 0 0 ▼	4 0 0 ▼	8 0 0 ▼	8 12 4 △
8	8 9 0 △	8 9 0 △	8 12 0 △	8 9 0 △	8 12 6 △	8 1 0 ▼	8 2 0 ▼	8 12 6 △	8 4 0 ▼
9	8 12 2 △	8 12 2 △	7 0 0 ▼	8 12 1 △	8 1 0 ▼	4 0 0 ▼	8 12 6 △	8 0 0 ▼	8 6 0 △
10	8 1 0 ▼	8 12 5 △	8 0 0 ▼	8 12 0 △	8 0 0 ▼	8 4 0 ▼	8 12 4 △	6 0 0 ▼	8 12 6 △
11	8 5 0 =	8 0 0 ▼	8 2 0 ▼	8 12 4 △	8 6 0 △	2 0 0 ▼	8 12 0 △	8 12 5 △	8 2 0 ▼
12	8 3 0 ▼	8 8 0 △	5 0 0 ▼	8 12 3 △	8 0 0 ▼	8 12 1 △	8 10 0 △	8 12 1 △	8 12 0 △
13	8 6 0 △	8 11 0 △	lock ▼	8 1 0 ▼	8 0 0 ▼	8 12 5 △	8 11 0 △	8 12 3 △	5 0 0 ▼
14	2 0 0 ▼	8 6 0 △	8 11 0 △	6 0 0 ▼	8 11 0 △	8 6 0 △	8 7 0 △	8 9 0 △	8 0 0 ▼
15	8 12 3 △	2 0 0 ▼	2 0 0 ▼	8 12 3 △	lock ▼	8 12 6 △	2 0 0 ▼	8 12 5 △	8 12 2 △
16	8 2 0 ▼	8 5 0 =	8 12 0 △	8 12 4 △	8 12 0 △	8 0 0 ▼	8 12 1 △	8 10 0 △	8 12 0 △
17	6 0 0 ▼	8 12 4 △	4 0 0 ▼	8 12 2 △	8 12 2 △	8 12 4 △	lock ▼	7 0 0 ▼	4 0 0 ▼
18	8 8 0 △	8 4 0 ▼	8 1 0 ▼	7 0 0 ▼	8 12 6 △	8 12 0 △	8 3 0 ▼	8 9 0 △	8 12 1 △
19	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
20	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
21	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
b=5	1	2	3	4	5	6	7	8	9
1	8 12 3 △	5 0 0 ▼	8 2 0 ▼	8 1 0 ▼	8 12 1 △	8 9 0 △	8 3 0 ▼	8 4 0 =	8 12 2 △
2	8 1 0 ▼	8 2 0 ▼	8 8 0 △	8 0 0 ▼	8 9 0 △	5 0 0 ▼	8 12 5 △	8 12 4 △	8 12 6 △
3	8 12 3 △	1 0 0 ▼	8 0 0 ▼	8 9 0 △	8 12 1 △	8 12 1 △	8 11 0 △	6 0 0 ▼	8 4 0 =
4	8 12 0 △	8 12 4 △	3 0 0 ▼	8 12 5 △	8 1 0 ▼	8 11 0 △	4 0 0 ▼	8 4 0 =	8 0 0 ▼
5	8 5 0 △	8 2 0 ▼	8 8 0 △	8 7 0 △	8 12 0 △	8 2 0 ▼	8 5 0 △	7 0 0 ▼	8 5 0 △
6	lock ▼	5 0 0 ▼	8 12 5 △	8 7 0 △	8 1 0 ▼	4 0 0 ▼	8 0 0 ▼	5 0 0 ▼	2 0 0 ▼
7	8 5 0 △	3 0 0 ▼	8 2 0 ▼	8 4 0 =	8 9 0 △	8 12 4 △	8 3 0 ▼	2 0 0 ▼	8 4 0 =
8	8 8 0 △	8 12 4 △	8 9 0 △	8 8 0 △	3 0 0 ▼	8 12 2 △	8 9 0 △	8 2 0 ▼	8 12 2 △
9	8 12 5 △	8 12 3 △	8 10 0 △	8 9 0 △	8 11 0 △	lock ▼	lock ▼	8 12 5 △	3 0 0 ▼
10	8 0 0 ▼	1 0 0 ▼	1 0 0 ▼	8 9 0 △	8 12 5 △	6 0 0 ▼	8 7 0 △	8 12 2 △	8 8 0 △
11	8 10 0 △	3 0 0 ▼	8 7 0 △	8 11 0 △	8 9 0 △	7 0 0 ▼	6 0 0 ▼	8 8 0 △	8 8 0 △
12	8 11 0 △	8 12 5 △	8 12 0 △	8 1 0 ▼	1 0 0 ▼	8 8 0 △	8 12 6 △	8 10 0 △	8 4 0 =
13	8 1 0 ▼	3 0 0 ▼	lock ▼	6 0 0 ▼	8 11 0 △	8 12 6 △	8 5 0 △	8 9 0 △	8 12 3 △
14	8 7 0 △	lock ▼	6 0 0 ▼	8 7 0 △	8 5 0 △	8 5 0 △	8 4 0 =	8 0 0 ▼	8 12 6 △
15	8 12 4 △	3 0 0 ▼	lock ▼	4 0 0 ▼	8 12 5 △	8 12 4 △	1 0 0 ▼	3 0 0 ▼	8 9 0 △
16	2 0 0 ▼	5 0 0 ▼	lock ▼	8 12 6 △	8 1 0 ▼	8 9 0 △	8 12 1 △	8 7 0 △	8 12 5 △
17	8 12 3 △	1 0 0 ▼	8 8 0 △	lock ▼	3 0 0 ▼	8 8 0 △	7 0 0 ▼	1 0 0 ▼	8 12 5 △
18	2 0 0 ▼	3 0 0 ▼	7 0 0 ▼	8 9 0 △	8 7 0 △	8 12 4 △	8 0 0 ▼	8 10 0 △	8 6 0 △
19	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
20	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
21	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
b=4	1	2	3	4	5	6	7	8	9
1	1 0 0 ▼	1 0 0 ▼	8 12 3 △	8 4 0 △	lock ▼	8 12 4 △	5 0 0 ▼	8 12 3 △	8 6 0 △
2	8 12 6 △	lock ▼	7 0 0 ▼	8 6 0 △	8 12 2 △	8 0 0 ▼	8 12 0 △	8 11 0 △	8 4 0 △
3	8 6 0 △	8 7 0 △	8 12 1 △	8 1 0 ▼	2 0 0 ▼	4 0 0 ▼	8 5 0 △	4 0 0 ▼	8 6 0 △
4	8 12 4 △	2 0 0 ▼	8 10 0 △	8 5 0 △	8 6 0 △	8 12 5 △	8 9 0 △	lock ▼	8 1 0 ▼
5	8 12 6 △	8 12 4 △	8 5 0 △	8 5 0 △	8 12 0 △	8 0 0 ▼	8 12 1 △	8 12 2 △	8 12 3 △
6	8 10 0 △	8 6 0 △	8 2 0 ▼	5 0 0 ▼	8 9 0 △	8 4 0 △	4 0 0 ▼	8 12 4 △	8 12 0 △
7	8 11 0 △	8 0 0 ▼	1 0 0 ▼	lock ▼	8 12 6 △	8 12 5 △	8 12 5 △	8 11 0 △	5 0 0 ▼
8	8 4 0 △	8 3 0 =	7 0 0 ▼	8 11 0 △	8 6 0 △	8 6 0 △	8 6 0 △	7 0 0 ▼	8 12 0 △
9	8 12 4 △	5 0 0 ▼	8 12 0 △	8 9 0 △	8 12 3 △	8 8 0 △	lock ▼	8 12 6 △	4 0 0 ▼
10	8 0 0 ▼	lock ▼	8 10 0 △	2 0 0 ▼	8 8 0 △	8 8 0 △	1 0 0 ▼	1 0 0 ▼	8 7 0 △
11	8 1 0 ▼	1 0 0 ▼	8 12 0 △	8 8 0 △	8 8 0 △	8 12 6 △	8 8 0 △	8 10 0 △	8 11 0 △
12	8 10 0 △	3 0 0 ▼	8 12 6 △	8 12 3 △	3 0 0 ▼	8 1 0 ▼	8 7 0 △	8 11 0 △	8 12 6 △
13	8 5 0 △	5 0 0 ▼	8 5 0 △	1 0 0 ▼	8 12 0 △	8 0 0 ▼	8 2 0 ▼	8 1 0 ▼	6 0 0 ▼
14	8 6 0 △	8 6 0 △	8 3 0 =	8 12 0 △	8 10 0 △	8 12 5 △	8 12 3 △	8 12 4 △	8 0 0 ▼
15	3 0 0 ▼	8 12 1 △	7 0 0 ▼	8 10 0 △	4 0 0 ▼	lock ▼	8 0 0 ▼	6 0 0 ▼	8 6 0 △
16	2 0 0 ▼	8 9 0 △	8 2 0 ▼	8 11 0 △	8 4 0 △	8 12 6 △	8 12 2 △	8 2 0 ▼	8 12 5 △
17	8 2 0 ▼	8 12 0 △	lock ▼	7 0 0 ▼	8 4 0 △	8 12 1 △	8 12 6 △	8 8 0 △	5 0 0 ▼
18	8 12 5 △	6 0 0 ▼	8 11 0 △	8 12 3 △	6 0 0 ▼	1 0 0 ▼	8 0 0 ▼	8 12 6 △	8 9 0 △
19	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
20	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
21	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △

b=3	1	2	3	4	5	6	7	8	9
1	8 5 0 △	8 4 0 △	6 0 0 ▼	8 12 4 △	5 0 0 ▼	8 7 0 △	6 0 0 ▼	3 0 0 ▼	3 0 0 ▼
2	8 12 4 △	8 8 0 △	8 12 1 △	8 5 0 △	8 12 5 △	7 0 0 ▼	8 3 0 △	1 0 0 ▼	8 12 1 △
3	lock ▼	8 6 0 △	8 0 0 ▼	8 12 2 △	8 9 0 △	8 2 0 =	7 0 0 ▼	8 12 0 △	7 0 0 ▼
4	8 12 6 △	8 10 0 △	8 9 0 △	7 0 0 ▼	7 0 0 ▼	8 8 0 △	8 12 5 △	8 9 0 △	8 12 3 △
5	8 12 5 △	8 11 0 △	4 0 0 ▼	8 2 0 =	8 12 0 △	8 12 2 △	8 3 0 △	8 12 4 △	3 0 0 ▼
6	8 7 0 △	2 0 0 ▼	6 0 0 ▼	8 11 0 △	8 11 0 △	8 0 0 ▼	7 0 0 ▼	8 11 0 △	8 12 6 △
7	8 12 5 △	8 12 6 △	8 6 0 △	1 0 0 ▼	8 0 0 ▼	1 0 0 ▼	8 12 2 △	8 12 0 △	8 6 0 △
8	8 12 5 △	8 12 1 △	8 0 0 ▼	8 9 0 △	7 0 0 ▼	8 8 0 △	8 12 0 △	6 0 0 ▼	8 9 0 △
9	8 5 0 △	8 9 0 △	8 2 0 =	4 0 0 ▼	8 8 0 △	8 9 0 △	8 12 3 △	8 10 0 △	7 0 0 ▼
10	8 4 0 △	lock ▼	8 2 0 =	6 0 0 ▼	8 3 0 △	3 0 0 ▼	8 5 0 △	8 1 0 ▼	8 4 0 △
11	2 0 0 ▼	8 2 0 =	8 8 0 △	8 3 0 △	8 6 0 △	8 1 0 ▼	4 0 0 ▼	8 4 0 △	8 2 0 =
12	8 5 0 △	8 12 1 △	8 11 0 △	2 0 0 ▼	8 11 0 △	8 12 2 △	8 12 0 △	8 7 0 △	8 12 5 △
13	8 12 3 △	8 6 0 △	8 3 0 △	4 0 0 ▼	8 6 0 △	8 12 1 △	8 12 0 △	lock ▼	8 12 1 △
14	8 12 4 △	8 7 0 △	8 12 6 △	8 0 0 ▼	8 2 0 =	1 0 0 ▼	8 6 0 △	8 7 0 △	8 12 6 △
15	8 12 6 △	4 0 0 ▼	8 8 0 △	8 8 0 △	8 12 6 △	lock ▼	4 0 0 ▼	8 4 0 △	8 12 3 △
16	8 2 0 =	8 9 0 △	8 10 0 △	8 12 6 △	8 3 0 △	4 0 0 ▼	8 0 0 ▼	8 4 0 △	8 9 0 △
17	8 3 0 △	8 6 0 △	8 8 0 △	8 3 0 △	8 12 0 △	8 5 0 △	8 12 4 △	4 0 0 ▼	8 12 4 △
18	8 10 0 △	4 0 0 ▼	8 12 4 △	1 0 0 ▼	8 6 0 △	3 0 0 ▼	3 0 0 ▼	8 3 0 △	8 12 4 △
19	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
20	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
21	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
b=2	1	2	3	4	5	6	7	8	9
1	lock ▼	8 12 3 △	8 10 0 △	4 0 0 ▼	8 5 0 △	8 12 3 △	8 6 0 △	8 0 0 ▼	8 9 0 △
2	8 2 0 △	8 7 0 △	7 0 0 ▼	3 0 0 ▼	8 12 6 △	8 12 6 △	5 0 0 ▼	8 10 0 △	8 12 4 △
3	8 3 0 △	6 0 0 ▼	8 12 3 △	8 6 0 △	8 4 0 △	8 12 3 △	8 3 0 △	8 12 3 △	8 12 3 △
4	8 0 0 ▼	8 8 0 △	8 12 4 △	8 0 0 ▼	lock ▼	8 7 0 △	8 12 4 △	8 3 0 △	5 0 0 ▼
5	8 12 5 △	8 12 6 △	8 10 0 △	8 4 0 △	4 0 0 ▼	8 1 0 =	8 2 0 △	1 0 0 ▼	8 12 2 △
6	8 11 0 △	8 5 0 △	5 0 0 ▼	8 10 0 △	8 1 0 =	8 12 0 △	8 10 0 △	4 0 0 ▼	8 8 0 △
7	1 0 0 ▼	1 0 0 ▼	8 9 0 △	8 6 0 △	3 0 0 ▼	1 0 0 ▼	7 0 0 ▼	8 9 0 △	8 4 0 △
8	8 7 0 △	6 0 0 ▼	3 0 0 ▼	lock ▼	8 9 0 △	2 0 0 ▼	8 12 2 △	8 12 1 △	8 12 3 △
9	8 12 1 △	8 12 6 △	2 0 0 ▼	5 0 0 ▼	8 4 0 △	8 11 0 △	8 12 5 △	8 9 0 △	1 0 0 ▼
10	8 12 0 △	8 12 6 △	3 0 0 ▼	8 6 0 △	3 0 0 ▼	7 0 0 ▼	8 11 0 △	8 1 0 =	lock ▼
11	8 12 3 △	5 0 0 ▼	4 0 0 ▼	8 12 6 △	2 0 0 ▼	4 0 0 ▼	8 12 6 △	8 1 0 =	4 0 0 ▼
12	1 0 0 ▼	3 0 0 ▼	8 5 0 △	5 0 0 ▼	8 12 1 △	8 1 0 =	8 12 3 △	8 12 3 △	8 6 0 △
13	lock ▼	8 4 0 △	8 5 0 △	8 12 4 △	8 3 0 △	3 0 0 ▼	8 12 5 △	8 12 2 △	7 0 0 ▼
14	8 11 0 △	lock ▼	2 0 0 ▼	5 0 0 ▼	8 12 0 △	8 12 6 △	8 12 3 △	8 12 1 △	8 8 0 △
15	lock ▼	4 0 0 ▼	8 0 0 ▼	8 12 3 △	8 12 1 △	4 0 0 ▼	8 8 0 △	8 12 5 △	8 7 0 △
16	1 0 0 ▼	4 0 0 ▼	8 2 0 △	8 12 6 △	8 10 0 △	1 0 0 ▼	8 5 0 △	8 3 0 △	4 0 0 ▼
17	2 0 0 ▼	1 0 0 ▼	6 0 0 ▼	8 8 0 △	7 0 0 ▼	8 5 0 △	6 0 0 ▼	8 5 0 △	8 10 0 △
18	8 6 0 △	6 0 0 ▼	3 0 0 ▼	8 2 0 △	8 0 0 ▼	8 0 0 ▼	8 3 0 △	8 12 3 △	8 5 0 △
19	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
20	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
21	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
b=1	1	2	3	4	5	6	7	8	9
1	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
2	1 0 0 =	lock =	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
3	8 0 0 =	8 12 5 △	8 12 6 △	8 12 1 △	8 12 3 △	8 12 4 △	8 5 0 △	8 12 6 △	8 12 4 △
4	7 0 0 =	4 0 0 =	8 10 0 △	8 12 5 △	8 12 2 △	2 0 0 =	8 7 0 △	4 0 0 =	8 7 0 △
5	8 0 0 =	lock =	8 4 0 △	1 0 0 =	8 4 0 △	8 0 0 =	8 1 0 △	lock =	8 12 4 △
6	4 0 0 =	8 4 0 △	8 12 2 △	8 4 0 △	8 0 0 =	6 0 0 =	8 7 0 △	8 12 0 △	8 5 0 △
7	8 5 0 △	1 0 0 =	8 10 0 △	1 0 0 =	8 12 6 △	8 12 5 △	8 8 0 △	8 5 0 △	8 4 0 △
8	8 0 0 =	6 0 0 =	8 12 6 △	1 0 0 =	8 4 0 △	8 12 1 △	8 12 2 △	8 12 6 △	4 0 0 =
9	6 0 0 =	lock =	lock =	lock =	8 12 5 △	2 0 0 =	8 1 0 △	8 3 0 △	8 9 0 △
10	8 0 0 =	6 0 0 =	3 0 0 =	5 0 0 =	8 3 0 △	8 11 0 △	8 9 0 △	8 5 0 △	8 9 0 △
11	4 0 0 =	2 0 0 =	8 4 0 △	8 7 0 △	8 2 0 △	8 12 2 △	8 0 0 =	8 11 0 △	7 0 0 =
12	8 0 0 =	3 0 0 =	lock =	1 0 0 =	8 12 2 △	6 0 0 =	8 7 0 △	8 5 0 △	8 10 0 △
13	6 0 0 =	4 0 0 =	6 0 0 =	lock =	8 12 1 △	8 12 0 △	8 5 0 △	8 0 0 =	8 12 5 △
14	5 0 0 =	1 0 0 =	5 0 0 =	8 6 0 △	5 0 0 =	2 0 0 =	4 0 0 =	8 0 0 =	8 5 0 △
15	lock =	2 0 0 =	3 0 0 =	8 1 0 △	8 6 0 △	3 0 0 =	8 1 0 △	8 7 0 △	8 5 0 △
16	lock =	8 1 0 △	8 4 0 △	7 0 0 =	8 2 0 △	5 0 0 =	lock =	8 8 0 △	8 12 2 △
17	4 0 0 =	8 1 0 △	3 0 0 =	8 1 0 △	1 0 0 =	8 12 1 △	1 0 0 =	5 0 0 =	lock =
18	8 1 0 △	8 7 0 △	8 4 0 △	8 3 0 △	6 0 0 =	8 4 0 △	8 12 6 △	3 0 0 =	7 0 0 =
19	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
20	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
21	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △

A.1.2 Compromise alternative



A.1.3 Minimum hydropower potential deficit



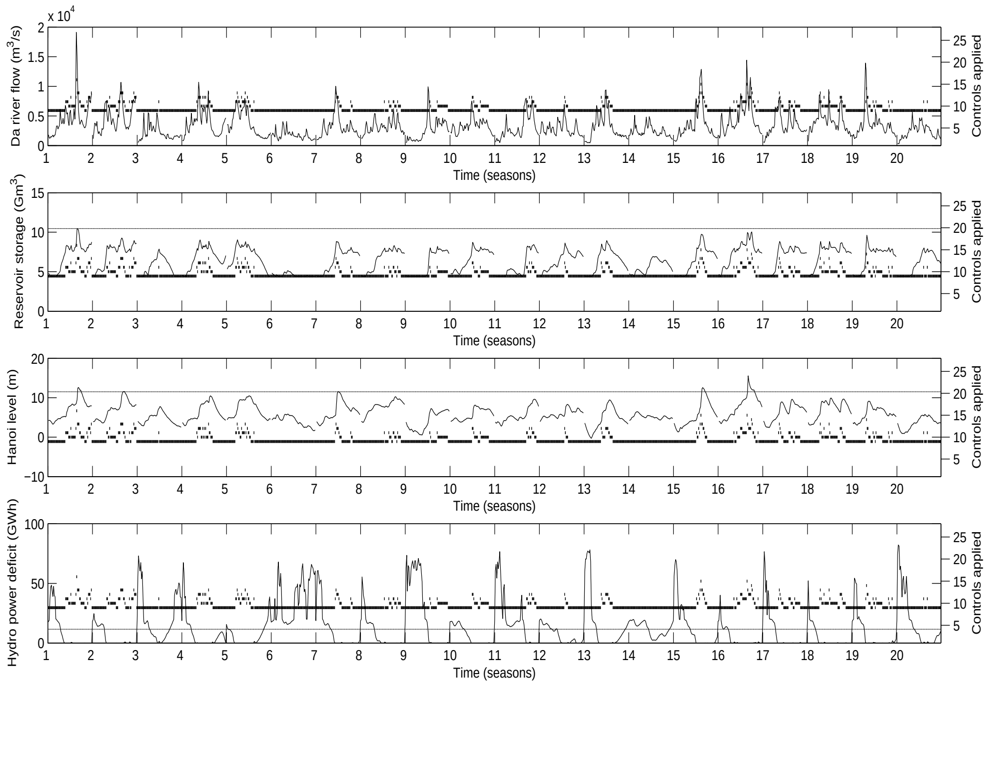
A.2 Optimization policies

The following tables and simulations describe the policies discussed in chapter 4.

An important aspect that can be noticed observing the simulations' plots is the effect of the purely stochastic description of inflows. In the series representing Hanoi levels, several flooding events (i.e.: level higher than +11.5 m) are registered for each policy. Such flooding events, applying the present control rule, have not occurred.

The reason of these apparently bad performances is, as stated above, the description of inflows used in the planning model. Applying a purely stochastic model, events like several days of continuously high inflows, although registered in the historical time-series used for calibration, have a minimal probability to happen. Any optimal policy obtained on this basis gives, consequently, a minimal weight to this kind of events. However, the expectation for this kind of results motivated the proposal of the POLFC described in section 3.2.7. Keeping into account the self-correlation of inflows, this approach will compensate these lacks of the planning model, improving the resulting policies.

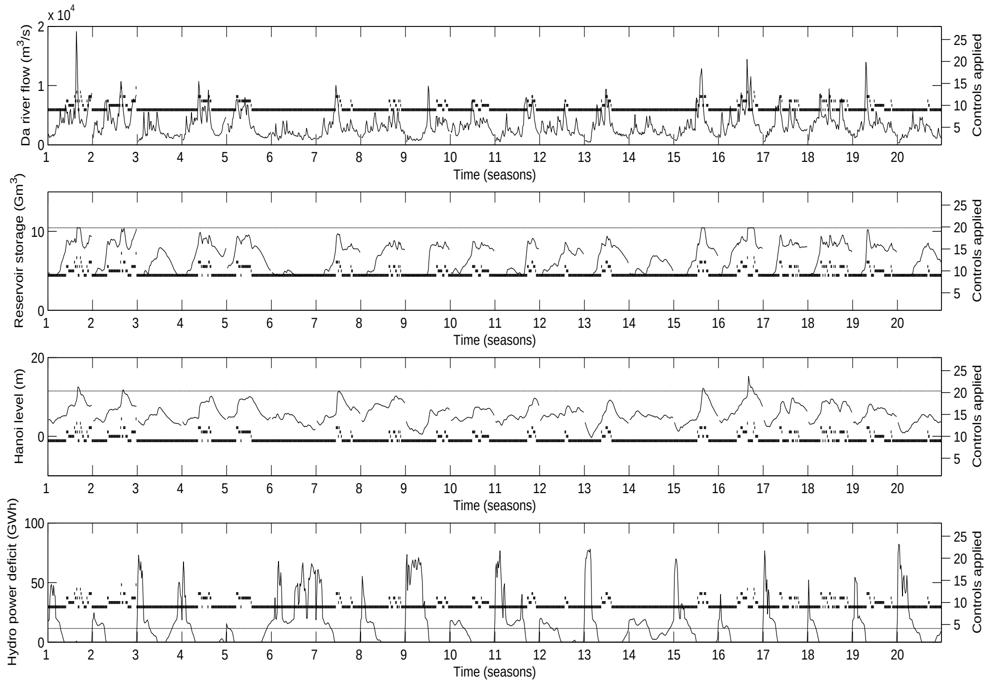
A.2.1 Minimum downstream flooding



b=7	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼
12	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼
13	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 3 0 ▼	8 2 0 ▼
14	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 2 0 ▼	8 3 0 ▼	8 2 0 ▼
15	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 3 0 ▼	8 2 0 ▼
16	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
17	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
18	8 5 0 ▼	8 5 0 ▼	8 5 0 ▼	8 5 0 ▼	8 5 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
19	8 5 0 ▼	8 5 0 ▼	8 5 0 ▼	8 5 0 ▼	8 5 0 ▼	8 5 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼
20	8 6 0 =	8 6 0 =	8 6 0 =	8 6 0 =	8 6 0 =	8 5 0 ▼	8 5 0 ▼	8 4 0 ▼	8 3 0 ▼
21	8 7 0 =	8 7 0 =	8 7 0 =	8 7 0 =	8 7 0 =	8 6 0 =	8 5 0 ▼	8 4 0 ▼	8 4 0 ▼
b=6	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
6	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 1 0 ▼
12	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 1 0 ▼
13	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 2 0 ▼	8 3 0 ▼	8 1 0 ▼
14	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼	8 3 0 ▼	8 1 0 ▼
15	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 1 0 ▼
16	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼
17	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 3 0 ▼	8 3 0 ▼
18	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 4 0 ▼	8 4 0 ▼	8 3 0 ▼	8 3 0 ▼
19	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 4 0 ▼	8 3 0 ▼
20	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 4 0 ▼	8 3 0 ▼
21	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 4 0 ▼	8 3 0 ▼
b=5	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 0 0 ▼
12	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 0 0 ▼
13	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼
14	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 1 0 ▼
15	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼
16	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 4 0 =	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼
17	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 3 0 ▼	8 2 0 ▼
18	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 3 0 ▼
19	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 4 0 =	8 4 0 =	8 3 0 ▼
20	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 4 0 =	8 4 0 =	8 4 0 =
21	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 6 0 △	8 4 0 =	8 4 0 =	8 4 0 =
b=4	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼
12	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
13	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
14	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 1 0 ▼	8 1 0 ▼
15	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 1 0 ▼	8 1 0 ▼
16	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =
17	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 3 0 =	8 3 0 =	8 3 0 =
18	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 3 0 =	8 3 0 =
19	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 5 0 △	8 3 0 =	8 3 0 =
20	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 5 0 △	8 3 0 =
21	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △

b=3	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
12	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼	8 0 0 ▼
13	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 2 0 =	8 2 0 =	8 0 0 ▼	8 0 0 ▼
14	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 0 0 ▼	8 2 0 =
15	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =
16	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 2 0 =	8 2 0 =
17	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 2 0 =	8 2 0 =
18	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 4 0 △	8 4 0 △	8 2 0 =
19	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 4 0 △	8 4 0 △	8 4 0 △
20	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 4 0 △	8 4 0 △	8 4 0 △
21	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 5 0 △	8 4 0 △	8 4 0 △
b=2	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
12	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 0 0 ▼	8 0 0 ▼
13	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 0 0 ▼	8 0 0 ▼
14	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =	8 1 0 =	8 0 0 ▼
15	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =	8 1 0 =
16	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =
17	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 3 0 △	8 3 0 △	8 1 0 =
18	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 4 0 △	8 3 0 △	8 3 0 △
19	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 5 0 △	8 3 0 △	8 3 0 △
20	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 4 0 △	8 4 0 △	8 3 0 △
21	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 7 0 △	8 5 0 △	8 4 0 △	8 3 0 △
b=1	1	2	3	4	5	6	7	8	9
1	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
2	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
3	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
4	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
5	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
6	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
7	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
8	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
9	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
10	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
11	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
12	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
13	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 0 0 =	8 0 0 =
14	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 0 0 =	8 0 0 =
15	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 2 0 △	8 2 0 △	8 12 6 △
16	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 3 0 △	8 2 0 △	8 12 6 △
17	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 4 0 △	8 2 0 △	8 12 5 △
18	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 4 0 △	8 2 0 △	8 12 5 △
19	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 4 0 △	8 12 1 △	8 12 5 △
20	8 8 0 △	8 8 0 △	8 8 0 △	8 8 0 △	8 8 0 △	8 8 0 △	8 5 0 △	8 12 1 △	8 12 5 △
21	8 8 0 △	8 8 0 △	8 8 0 △	8 8 0 △	8 8 0 △	8 8 0 △	8 6 0 △	8 12 2 △	8 12 5 △

A.2.2 Minimum hydropower deficit



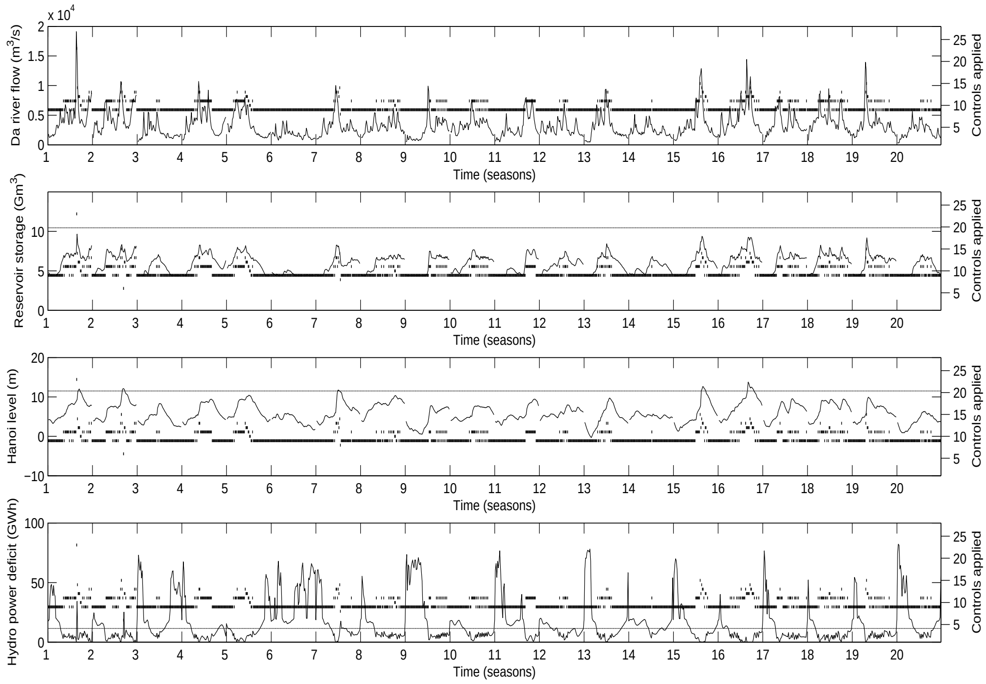
b=7	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
12	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
13	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
14	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
15	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼
16	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼
17	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼
18	8 0 0 ▼	8 2 0 ▼	8 3 0 ▼	8 2 0 ▼	8 3 0 ▼	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 2 0 ▼
19	8 1 0 ▼	8 2 0 ▼	8 3 0 ▼	8 2 0 ▼	8 1 0 ▼	8 2 0 ▼	8 3 0 ▼	8 4 0 ▼	8 2 0 ▼
20	8 1 0 ▼	8 3 0 ▼	8 4 0 ▼	8 4 0 ▼	8 3 0 ▼	8 1 0 ▼	8 3 0 ▼	8 4 0 ▼	8 2 0 ▼
21	8 1 0 ▼	8 2 0 ▼	8 4 0 ▼	8 4 0 ▼	8 3 0 ▼	8 1 0 ▼	8 3 0 ▼	8 4 0 ▼	8 2 0 ▼
b=6	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
6	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
12	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
13	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
14	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼
15	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼
16	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼	8 1 0 ▼
17	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼	8 1 0 ▼
18	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼	8 3 0 ▼	8 3 0 ▼	8 1 0 ▼
19	8 1 0 ▼	8 3 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 1 0 ▼	8 3 0 ▼	8 3 0 ▼	8 1 0 ▼
20	8 1 0 ▼	8 3 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 1 0 ▼	8 3 0 ▼	8 3 0 ▼	8 1 0 ▼
21	8 3 0 ▼	8 3 0 ▼	8 5 0 =	8 3 0 ▼	8 3 0 ▼	8 1 0 ▼	8 3 0 ▼	8 3 0 ▼	8 1 0 ▼
b=5	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
12	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
13	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
14	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
15	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
16	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
17	8 2 0 ▼	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
18	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
19	8 3 0 ▼	8 3 0 ▼	8 4 0 =	8 4 0 =	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼
20	8 3 0 ▼	8 4 0 =	8 4 0 =	8 4 0 =	8 4 0 =	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
21	8 3 0 ▼	8 4 0 =	8 4 0 =	8 4 0 =	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
b=4	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
12	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
13	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
14	8 1 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
15	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
16	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
17	8 2 0 ▼	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
18	8 3 0 =	8 3 0 =	8 3 0 =	8 2 0 ▼	8 1 0 ▼	8 3 0 =	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼
19	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
20	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
21	8 3 0 =	8 3 0 =	8 5 0 △	8 3 0 =	8 3 0 =	8 3 0 =	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼

b=3	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
12	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
13	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
14	8 1 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
15	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
16	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
17	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 1 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
18	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 0 0 ▼	8 0 0 ▼
19	8 2 0 =	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
20	8 2 0 =	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 2 0 =	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
21	8 5 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼

b=2	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
11	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
12	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼
13	8 1 0 =	8 0 0 ▼	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 0 0 ▼	8 0 0 ▼
14	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 0 0 ▼	8 0 0 ▼
15	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 0 0 ▼	8 0 0 ▼
16	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 0 0 ▼	8 0 0 ▼
17	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =	8 3 0 △	8 1 0 =	8 0 0 ▼	8 0 0 ▼
18	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =	8 1 0 =	8 0 0 ▼
19	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =	8 1 0 =	8 0 0 ▼
20	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =	8 1 0 =	8 0 0 ▼
21	8 3 0 △	8 3 0 △	8 5 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =	8 1 0 =	8 0 0 ▼

b=1	1	2	3	4	5	6	7	8	9
1	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
2	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
3	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
4	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
5	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
6	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
7	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
8	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
9	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
10	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
11	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
12	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
13	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
14	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =
15	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 0 0 =	8 0 0 =	8 0 0 =
16	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 0 0 =	8 0 0 =	8 0 0 =
17	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 2 0 △	8 3 0 △	8 0 0 =	8 0 0 =	8 0 0 =
18	8 3 0 △	8 3 0 △	8 4 0 △	8 2 0 △	8 2 0 △	8 3 0 △	8 0 0 =	8 0 0 =	8 0 0 =
19	8 3 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 3 0 △	8 3 0 △	8 0 0 =	8 0 0 =	8 0 0 =
20	8 3 0 △	8 3 0 △	8 4 0 △	8 2 0 △	8 2 0 △	8 3 0 △	8 0 0 =	8 0 0 =	8 0 0 =
21	8 3 0 △	8 4 0 △	8 4 0 △	8 2 0 △	8 2 0 △	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =

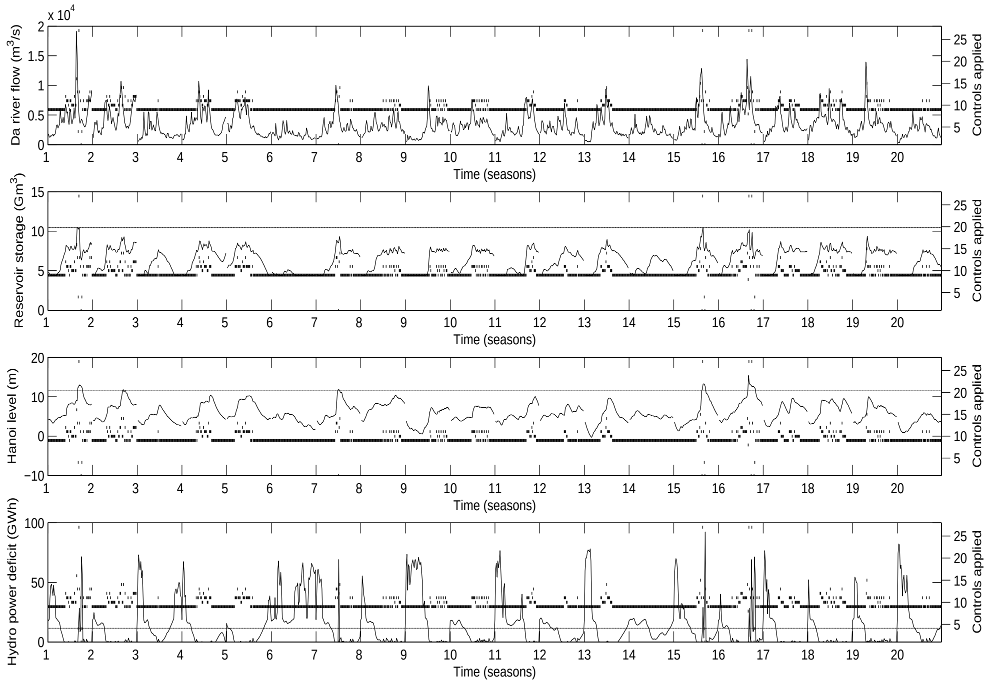
A.2.3 Minimum upstream flooding



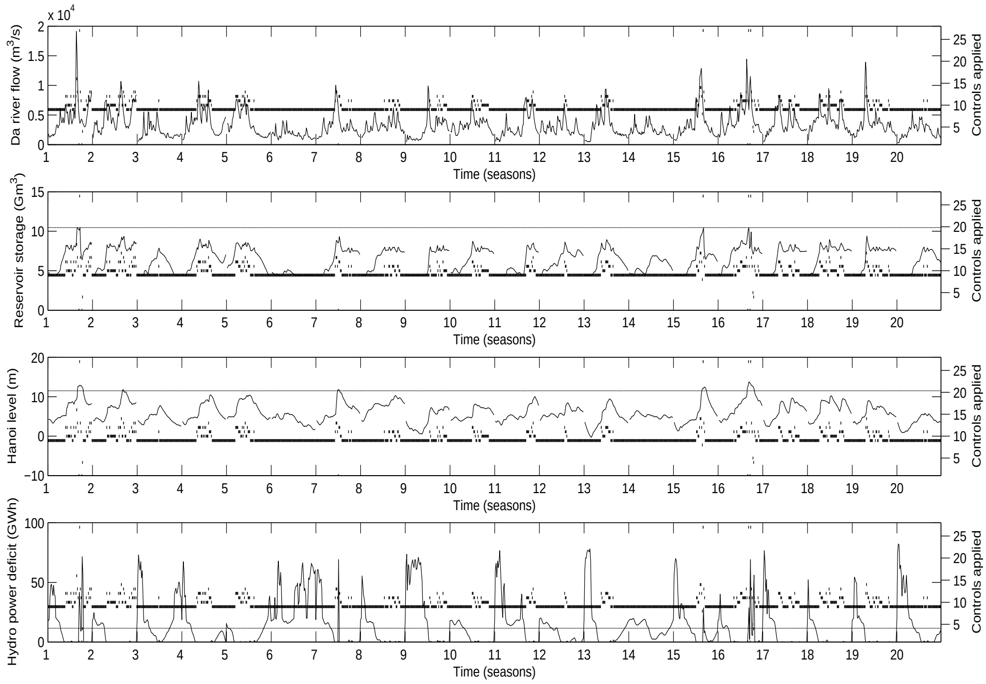
b=7	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 4 0 ▼	8 2 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 4 0 ▼	8 3 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 4 0 ▼	8 2 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 0 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
9	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼	8 1 0 ▼	8 3 0 ▼	8 0 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
10	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼	8 5 0 ▼	8 0 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
11	8 0 0 ▼	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
12	8 1 0 ▼	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
13	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 5 0 ▼	8 5 0 ▼	8 3 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
14	8 3 0 ▼	8 3 0 ▼	8 4 0 ▼	8 5 0 ▼	8 4 0 ▼	8 3 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
15	8 4 0 ▼	8 4 0 ▼	8 7 0 =	8 5 0 ▼	8 5 0 ▼	8 3 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
16	8 6 0 =	8 6 0 =	8 7 0 =	8 6 0 =	8 5 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 2 0 ▼
17	8 9 0 =	8 9 0 =	8 7 0 =	8 6 0 =	8 5 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼
18	8 12 2 =	8 12 0 =	8 10 0 =	8 8 0 =	8 6 0 =	8 5 0 ▼	8 5 0 ▼	8 4 0 ▼	8 4 0 ▼
19	8 12 3 =	8 12 2 =	8 12 0 =	8 10 0 =	8 8 0 =	8 7 0 =	8 5 0 ▼	8 5 0 ▼	8 5 0 ▼
20	8 12 5 =	8 12 3 =	8 12 2 =	8 12 0 =	8 10 0 =	8 9 0 =	8 7 0 =	8 6 0 =	8 6 0 =
21	8 12 6 =	8 12 5 =	8 12 4 =	8 12 3 =	8 12 1 =	8 12 2 =	8 11 0 =	8 12 2 =	8 12 6 =
b=6	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 1 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 1 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 3 0 ▼	8 2 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
6	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	7 0 0 ▼	8 1 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 3 0 ▼	8 2 0 ▼	8 3 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 4 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
10	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼	8 5 0 =	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	7 0 0 ▼
11	8 1 0 ▼	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 5 0 =	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
12	8 2 0 ▼	8 2 0 ▼	8 3 0 ▼	8 5 0 =	8 5 0 =	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
13	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 5 0 =	8 3 0 ▼	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
14	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 5 0 =	8 5 0 =	8 4 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
15	8 4 0 ▼	8 4 0 ▼	8 4 0 ▼	8 5 0 =	8 5 0 =	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼	8 0 0 ▼
16	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 4 0 ▼	8 3 0 ▼	8 3 0 ▼	8 3 0 ▼
17	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 4 0 ▼	8 4 0 ▼	8 3 0 ▼
18	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 4 0 ▼	8 4 0 ▼
19	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =
20	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =
21	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =	8 5 0 =
b=5	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 12 6 △	8 12 5 △
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼	8 12 6 △
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 4 0 =	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
10	8 1 0 ▼	8 2 0 ▼	8 1 0 ▼	8 2 0 ▼	8 4 0 =	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
11	8 2 0 ▼	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 4 0 =	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
12	8 2 0 ▼	8 2 0 ▼	8 3 0 ▼	8 3 0 ▼	8 4 0 =	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼	7 0 0 ▼
13	8 3 0 ▼	8 2 0 ▼	8 3 0 ▼	8 4 0 =	8 4 0 =	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	3 0 0 ▼
14	8 4 0 =	8 4 0 =	8 6 0 △	8 4 0 =	8 4 0 =	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 0 0 ▼
15	8 6 0 △	8 6 0 △	8 7 0 △	8 6 0 △	8 4 0 =	8 4 0 =	8 3 0 ▼	8 2 0 ▼	8 2 0 ▼
16	8 6 0 △	8 7 0 △	8 7 0 △	8 6 0 △	8 6 0 △	8 4 0 =	8 4 0 =	8 4 0 =	8 2 0 ▼
17	8 6 0 △	8 6 0 △	8 7 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 4 0 =	8 4 0 =	8 4 0 =
18	8 7 0 △	8 9 0 △	8 7 0 △	8 7 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 4 0 =	8 4 0 =
19	8 8 0 △	8 9 0 △	8 8 0 △	8 7 0 △	8 7 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △
20	8 11 0 △	8 9 0 △	8 9 0 △	8 8 0 △	8 7 0 △	8 7 0 △	8 6 0 △	8 6 0 △	8 6 0 △
21	8 12 0 △	8 11 0 △	8 10 0 △	8 9 0 △	8 8 0 △	8 7 0 △	8 7 0 △	8 6 0 △	8 6 0 △
b=4	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼	8 0 0 ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼	8 0 0 ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 0 0 ▼	8 0 0 ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 3 0 =	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 ▼	8 3 0 =	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
9	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 2 0 ▼	8 2 0 ▼	8 3 0 =	8 1 0 ▼	8 1 0 ▼	7 0 0 ▼
10	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 2 0 ▼	8 5 0 △	8 3 0 =	8 1 0 ▼	8 1 0 ▼	8 0 0 ▼
11	8 2 0 ▼	8 2 0 ▼	8 3 0 =	8 3 0 =	8 5 0 △	8 3 0 =	8 1 0 ▼	8 1 0 ▼	7 0 0 ▼
12	8 2 0 ▼	8 3 0 =	8 3 0 =	8 5 0 △	8 5 0 △	8 3 0 =	8 3 0 =	8 1 0 ▼	7 0 0 ▼
13	8 3 0 =	8 3 0 =	8 3 0 =	8 5 0 △	8 5 0 △	8 3 0 =	8 3 0 =	8 1 0 ▼	7 0 0 ▼
14	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 3 0 =	8 3 0 =	8 1 0 ▼	8 0 0 ▼
15	8 6 0 △	8 6 0 △	8 7 0 △	8 5 0 △	8 5 0 △	8 3 0 =	8 3 0 =	8 3 0 =	8 3 0 =
16	8 6 0 △	8 6 0 △	8 7 0 △	8 6 0 △	8 5 0 △	8 5 0 △	8 3 0 =	8 3 0 =	8 3 0 =
17	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 3 0 =
18	8 8 0 △	8 8 0 △	8 7 0 △	8 7 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 5 0 △	8 5 0 △
19	8 12 2 △	8 8 0 △	8 8 0 △	8 8 0 △	8 7 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 5 0 △
20	8 12 2 △	8 9 0 △	8 9 0 △	8 8 0 △	8 7 0 △	8 7 0 △	8 6 0 △	8 6 0 △	8 6 0 △
21	8 12 2 △	8 12 0 △	8 11 0 △	8 9 0 △	8 8 0 △	8 7 0 △	8 7 0 △	8 5 0 △	8 5 0 △

b=3	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	lock ▼	lock ▼	lock ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	5 0 0 ▼	2 0 0 ▼	lock ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	3 0 0 ▼	1 0 0 ▼	lock ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	2 0 0 ▼	lock ▼	lock ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	lock ▼	lock ▼	lock ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	lock ▼	lock ▼	lock ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 1 0 ▼	lock ▼	lock ▼	lock ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	lock ▼	lock ▼
9	8 0 0 ▼	8 0 0 ▼	8 1 0 ▼	8 2 0 =	8 2 0 =	8 1 0 ▼	8 0 0 ▼	4 0 0 ▼	1 0 0 ▼
10	8 2 0 =	8 2 0 =	8 2 0 =	8 2 0 =	8 4 0 △	8 2 0 =	6 0 0 ▼	4 0 0 ▼	lock ▼
11	8 2 0 =	8 2 0 =	8 2 0 =	8 5 0 △	8 5 0 △	8 2 0 =	5 0 0 ▼	2 0 0 ▼	lock ▼
12	8 2 0 =	8 2 0 =	8 4 0 △	8 5 0 △	8 4 0 △	8 2 0 =	8 0 0 ▼	3 0 0 ▼	lock ▼
13	8 4 0 △	8 4 0 △	8 5 0 △	8 5 0 △	8 4 0 △	8 2 0 =	8 2 0 =	8 0 0 ▼	lock ▼
14	8 4 0 △	8 4 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 4 0 △	8 2 0 =	8 2 0 =	8 0 0 ▼
15	8 6 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 5 0 △	8 4 0 △	8 4 0 △	8 2 0 =	8 2 0 =
16	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 6 0 △	8 4 0 △	8 4 0 △	8 4 0 △	8 4 0 △
17	8 6 0 △	8 9 0 △	8 9 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 4 0 △	8 4 0 △	8 4 0 △
18	8 12 1 △	8 12 2 △	8 8 0 △	8 8 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 5 0 △	8 4 0 △
19	8 12 1 △	8 12 2 △	8 11 0 △	8 9 0 △	8 8 0 △	8 6 0 △	8 6 0 △	8 5 0 △	8 5 0 △
20	8 12 2 △	8 12 2 △	8 11 0 △	8 12 0 △	8 9 0 △	8 8 0 △	8 6 0 △	8 6 0 △	8 6 0 △
21	8 12 6 △	8 12 2 △	8 12 0 △	8 12 0 △	8 10 0 △	8 8 0 △	8 8 0 △	8 7 0 △	8 12 2 △
b=2	1	2	3	4	5	6	7	8	9
1	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	5 0 0 ▼	lock ▼	lock ▼
2	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	2 0 0 ▼	1 0 0 ▼
3	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	1 0 0 ▼	lock ▼
4	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	1 0 0 ▼	lock ▼
5	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	1 0 0 ▼	lock ▼
6	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 =	8 0 0 ▼	lock ▼	lock ▼
7	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 =	8 3 0 △	8 0 0 ▼	lock ▼	1 0 0 ▼
8	8 0 0 ▼	8 0 0 ▼	8 0 0 ▼	8 1 0 =	8 1 0 =	8 3 0 △	8 0 0 ▼	lock ▼	lock ▼
9	8 1 0 =	8 1 0 =	8 1 0 =	8 1 0 =	8 3 0 △	8 3 0 △	8 0 0 ▼	lock ▼	lock ▼
10	8 1 0 =	8 1 0 =	8 1 0 =	8 3 0 △	8 4 0 △	8 3 0 △	8 0 0 ▼	lock ▼	lock ▼
11	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =	4 0 0 ▼	lock ▼
12	8 3 0 △	8 3 0 △	8 3 0 △	8 5 0 △	8 4 0 △	8 3 0 △	8 1 0 =	6 0 0 ▼	4 0 0 ▼
13	8 3 0 △	8 3 0 △	8 3 0 △	8 6 0 △	8 4 0 △	8 3 0 △	8 1 0 =	8 1 0 =	5 0 0 ▼
14	8 4 0 △	8 4 0 △	8 5 0 △	8 5 0 △	8 5 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 1 0 =
15	8 6 0 △	8 7 0 △	8 12 4 △	8 6 0 △	8 5 0 △	8 4 0 △	8 3 0 △	8 3 0 △	8 3 0 △
16	8 6 0 △	8 11 0 △	8 12 3 △	8 6 0 △	8 6 0 △	8 5 0 △	8 4 0 △	8 3 0 △	8 3 0 △
17	8 9 0 △	8 12 1 △	8 12 3 △	8 7 0 △	8 7 0 △	8 5 0 △	8 5 0 △	8 4 0 △	8 3 0 △
18	8 12 3 △	8 12 6 △	8 12 3 △	8 11 0 △	8 7 0 △	8 6 0 △	8 5 0 △	8 5 0 △	8 12 6 △
19	8 12 4 △	8 12 6 △	8 12 2 △	8 12 2 △	8 12 5 △	8 12 5 △	8 12 5 △	8 12 6 △	8 12 6 △
20	8 12 4 △	8 12 6 △	8 12 6 △	8 12 3 △	8 12 5 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
21	8 12 6 △	8 12 6 △	8 12 5 △	8 12 4 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
b=1	1	2	3	4	5	6	7	8	9
1	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	3 0 0 =	lock =	lock =
2	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	3 0 0 =	lock =	lock =
3	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	3 0 0 =	lock =	lock =
4	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	3 0 0 =	lock =	lock =
5	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	3 0 0 =	lock =	lock =
6	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 3 0 △	3 0 0 =	lock =	lock =
7	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 4 0 △	3 0 0 =	lock =	lock =
8	8 0 0 =	8 0 0 =	8 0 0 =	8 0 0 =	8 2 0 △	8 3 0 △	3 0 0 =	lock =	lock =
9	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 2 0 △	8 3 0 △	3 0 0 =	lock =	lock =
10	8 2 0 △	8 2 0 △	8 3 0 △	8 2 0 △	8 3 0 △	8 3 0 △	3 0 0 =	lock =	lock =
11	8 2 0 △	8 2 0 △	8 3 0 △	8 3 0 △	8 3 0 △	8 3 0 △	4 0 0 =	lock =	lock =
12	8 2 0 △	8 2 0 △	8 3 0 △	8 5 0 △	8 3 0 △	8 3 0 △	8 2 0 △	4 0 0 =	8 12 6 △
13	8 3 0 △	8 3 0 △	8 3 0 △	8 5 0 △	8 5 0 △	8 3 0 △	8 2 0 △	8 12 6 △	8 12 6 △
14	8 4 0 △	8 4 0 △	8 4 0 △	8 9 0 △	8 4 0 △	8 3 0 △	8 12 6 △	8 12 6 △	8 12 6 △
15	8 6 0 △	8 7 0 △	8 12 6 △	8 12 6 △	8 5 0 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
16	8 6 0 △	8 6 0 △	8 12 5 △	8 12 6 △	8 6 0 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
17	8 9 0 △	8 12 1 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
18	8 12 3 △	8 12 0 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
19	8 12 3 △	8 12 1 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
20	8 12 2 △	8 12 1 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △
21	8 12 4 △	8 12 5 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △	8 12 6 △

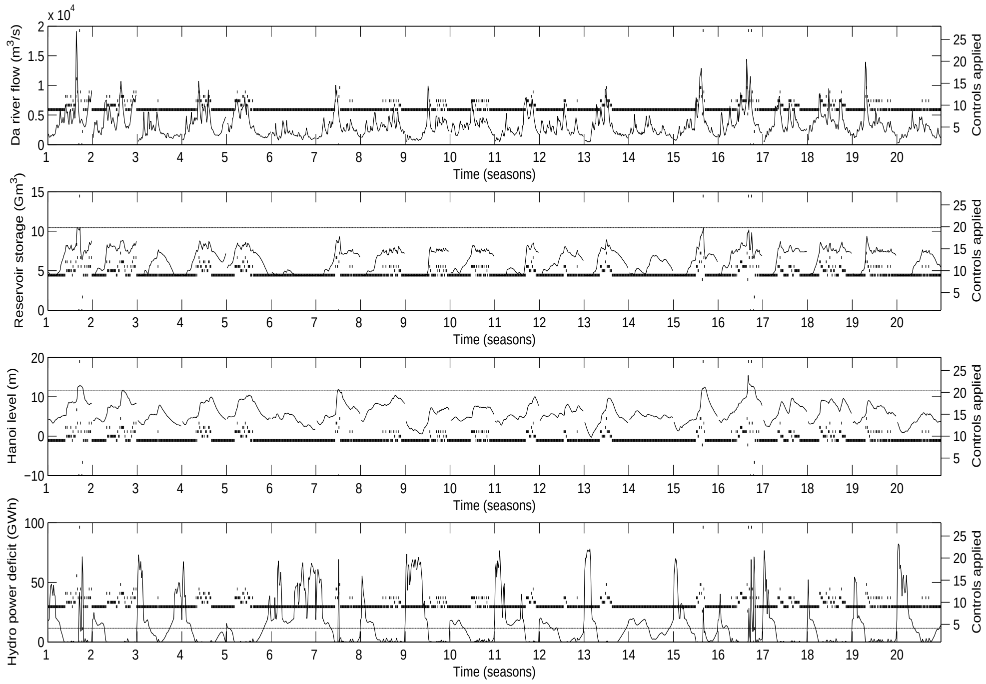
A.2.4 P3


x 10^4
Da river flow (m^3/s)
Controls applied
Time (seasons)
Reservoir storage (Gm^3)
Controls applied
Time (seasons)
Hanoi level (m)
Controls applied
Time (seasons)
Hydro power deficit (GWh)
Controls applied
Time (seasons)

A.2.5 P4


x 10^4
Da river flow (m^3/s)
Controls applied
Time (seasons)
Reservoir storage (Gm^3)
Controls applied
Time (seasons)
Hanoi level (m)
Controls applied
Time (seasons)
Hydro power deficit (GWh)
Controls applied
Time (seasons)

A.2.6 P5


x 10^4
Da river flow (m^3/s)
Reservoir storage (Gm^3)
Hanoi level (m)
Hydro power deficit (GWh)
Time (seasons)
Controls applied

Bibliography

Arnold, E., Tatjewski, P., Wolochowicz, P., 1994. Two methods for large-scale nonlinear optimization and their comparison on a case study of hydropower optimization. Journal of Optimization Theory Applications 81 (2), 221–248.

Back, T., Fogel, D. B., Michalewicz, Z. (Eds.), 1997. Handbook of Evolutionary Computation. Bristol, New York: Institute of Physics Publishing and Oxford University Press.

Barros, M., Tsai, F., Yang, S., Yeh, W., 2003. Optimization of large-scale hydropower system operations. Journal of water resources planning and management 129 (3), 178–188.

Bazarra, M., Sherali, H., Shetty, C., 1993. Nonlinear programming: theory and algorithms. Wiley, New York.

Bertsekas, D. P., 1995. Dynamic Programming and Optimal Control. Athena Scientific.

Bertsekas, D. P., Shreve, S. E., 1978. Stochastic optimal control. Academic Press, NY-S.Francisco-London.

Bertsekas, D. P., Tsitsiklis, J., 1996. Neuro-dynamic programming. Athena scientific, Belmont, Mass.

CCFSC, 2005. Central Committee for Flood and Storm Control, regulation on operation of hoa binh hydroelectric power reservoir and flood cutting constructions for red river in flood seasons, in vietnamese.

Crawley, P., Dandy, G., 1993. Optimal operation of multiple-reservoir system. Journal of water resources planning and management 119 (1), 1–17.

De Rigo, D., Rizzoli, A. E., Soncini Sessa, R., Weber, E., Zenesi, P., 2001. Neuro-dynamic programming for the efficient management of reservoir networks. In: MODSIM 2001 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand.

DHI, 2005. MIKE 11 - A modelling system for Rivers and channels. DHI Water and Environment, Denmark.

Dorini, G., 2006. The neighbour search approach for solving multi-objective markov decision processes, and the application in reservoirs operation planning. Ph.D. thesis, University of Exeter, to be submitted in December 2006.

Dorini, G., Di Pierro, F., Piunovskiy, A. B., Savic, D., 2006a. The neighbourhood search for constructing pareto sets. Math Methods Oper. Rese.Accepted, August 2006.

Dorini, G., Jonkergouw, P., Kapelan, Z., di Pierro, F., Khu, S., Savic, D., 2006b. An efficient algorithm for sensor placement in water distribution systems. In: Proceedings, 8th Water Distribution Systems Analysis Symposium. Cincinnati, Ohio, USA.

Dynkin, E. B., Yushkevich, A. A., 1979. Controlled Markov Process and their applications. Springer-Verlag New York.

Esat, V., Hall, M., 1994. Water resources system optimisation using genetic algorithms. In: Proceedings, 1st International Conference on Hydroinformatics, Balkema. pp. 225–231, rotterdam.

Eschenbach, E., Magee, T., Zagona, E., Goronfio, M., Shane, R., 2001. Goal programming decision support system for multi-objective operation of reservoir systems. Journal of water resources planning and management 127 (2), 108–120.

Feinberg, E., 2000. Constrained discounted markov decision processes and hamilton cycles. Math. of Oper. Res. (25), 130–140.

Feinberg, E., Shwartz, A., 1996. Constrained discounted dynamic programming. Math. of Oper. Res. (21), 922–945.

Georgakakos, A. P., 1989. Extended linear quadratic gaussian control for the real time operation of reservoir systems. In: Dynamic programming for optimal water resources systems analysis. Esogbue Editor, pp. 329–360.

Grygier, J., Stedinger, J., 1985. Algorithms for optimizing hydropower system operation. Water Resources Res. 21 (1), 1–10.

Hanne, T., 1999. On the convergence of multiobjective evolutionary algorithms. European Journal of Operational Research 117 (3), 553–564.

Hanne, T., 2001. Global multiobjective optimization with evolutionary algorithms: Selection mechanisms and mutation control. In: Springer, B. (Ed.), Proceedings of Evolutionary Multi-Criterion Optimization (EMO 2001). Vol. 1993. pp. 197–212.

Heyman, D., Sobel, M., 1994. Stochastic Models in Operational Research, Vol.II Stochastic Optimization. McGrow-Hill Book Company,NY.

Hiew, K., 1987. Optimization algorithms for large-scale multi-reservoir hydropower systems. Ph.D. thesis, Dept. of Civil Engineering, Colorado State University.

Hiew, K., Labadie, J., Scott, J., 1989. Optimal operation analysis of the Colorado-Big Thompson project. In: Labadie, J., et al. (Eds.), Computerized decision support systems for water managers. ASCE, pp. 632–646.

Huang, W., Harboe, R., Bogardi, J., 1991. Testing stochastic dynamic programming models conditioned on observed or forecasted inflows. Journal of water resources planning and management 117 (1), 28–36.

Jacobs, J., Freeman, G., et al., 1995. SOCRATES: A system for scheduling hydroelectric generation under uncertainty. In: Vladimirou, et al. (Eds.), Models for planning under uncertainty. Baltzer Science, Bussum, The Netherlands.

Jacobson, H., Mayne, Q., 1970. Differential dynamic programming. Elsevier, New York.

Jones, L., Willis, R., Finney, B., 1986. Water resources systems planning: differential dynamic programming models. In: Proceedings, Water Forum '86. pp. 1033–1040, ASCE, Reston, Va.

Kall, P., Wallace, S., 1995. Stochastic programming. Wiley, New York.

Ko, S.-K., Fontane, D., Labadie, J., 1992. Multi-objective optimization of reservoir systems operations. Water resources Bulletin 28 (1), 111–127.

Labadie, J., 1993. Combining simulation and optimization in river basin management. In: J., M., et al. (Eds.), Stochastic hydrology and its uses in water resources systems simulation and optimization. Kluwer Academic, pp. 345–371.

Labadie, J. W., 2004. Optimal operation of multireservoir systems: state-of-the-art review. Journal of water resources planning and management 130 (2), 93–111.

Lamond, B. F., Boukhtouta, A., 2002. Water reservoir applications of markov decision processes. In: Feinberg, E. A., Schwartz, A. (Eds.), Handbook of markov decision processes. Kluwer, Ch. 17, pp. 537–558.

Laumanns, M., 2003. Analysis and applications of evolutionary multiobjective optimization algorithms. Ph.D. thesis, Swiss Federal Institute of Technology Zurich.

Le Ngo, L., Madsen, H., Rosbjerg, D., Pedersen, C. B., 2006. Implementation and comparison of reservoir operation strategies for the Hoa Binh reservoir, Vietnam using the MIKE 11 model, water Resour. Management, Submitted.

Madsen, H., Le Ngo, L., Rosbjerg, D., 2006. Multi-objective optimisation of reservoir operation using surrogate modelling. In: 7th International conference on hydroinformatics. Nice, France.

McMullen, P., Sherphard, G. C., 1971. Convex polytopes and Upper Bound Conjecture. Cambridge University Press.

Piunovskiy, A. B., 1997. Optimal Control of Random Sequences in Problems with Constrains. Kluwer, Dordrecht.

Preparata, F., Shamos, M., 1993. Computational Geometry. Springer-Verlag,New York.

Rubinstein, R. Y., 1999. The simulated entropy method for combinatorial and continuous optimization. Methodology and Computing in Applied Probability 2, 127–190.

Rudolph, G., 1998. Finite markov chain results in evolutionary computation: A tour d'horizon. Fundamenta Informaticae 35, 67–89.

Seifi, A., Hipel, K., 2001. Interior-point method for reservoir operation with stochastic inflows. Journal of water resources planning and management 127 (1), 48–57.

Sharif, M., Wardlaw, R., 2000. Multireservoir systems optimisation using genetic algorithms: case study. Journal of Computing in Civil Engineering, ASCE 14 (4), 255–263.

Soncini Sessa, R., 2004. MODSS. Per decisioni integrate e partecipate. McGraw-Hill.

Stedinger, J. R., Sule, B. F., Loucks, D. P., 1984. Stochastic dynamic programming models for reservoir operation optimization. Water resources research 20, 1499–1505.

Tejada-Guibert, J., Johnson, S. A., Stedinger, J. R., 1993. Comparison of two approaches for implementing multi-reservoir operating policies derived using stochastic dynamic programming. Water resources research 29, 3969–3980.

Tejada-Guibert, J., Johnson, S. A., Stedinger, J. R., 1995. The value of hydrologic information in stochastic dynamic programming models of a multi-reservoir system. Water resources research 31, 2571–2579.

Terry, L., Pereira, M. V. F., Neto, T. A. A., Silva, L. F. C. A., Sales, P. R. H., 1986. Coordinating the energy generation of the brazilian national hydrothermal electrical generating system. Interfaces 16, 16–38.

Tinh, D. Q., 1999. Participatory planning and management for flood mitigation and preparedness and trends in the Red River basin, Viet Nam. Available from http://www.unescap.org/esd/water/disaster/2001/vietnam.doc in May 2006.

Unver, O., Mays, L., 1990. Model for real-time optimal flood control operation of a reservoir system. Water Resources Management 4, 21–46.

Vasiliadis, H., Karamouz, M., 1994. Demand-driven operation of reservoirs using uncertainty-based optimal operating policies. Journal of water resources planning and management 120 (1), 101–114.

WCD, 2000. World Commission on Dams. Dams and development: A new framework for decision-making. Earthscan Publications Ltd.

Wegman, E. J., 1990. Hyper-dimensional data analysis using parallel coordinates. Journal of American Statistical Association 85 (411), 664–675.

Wurbs, R. A., Tibbets, M. N., Cabezas, L. M., Roy, L. C., 1985. State-of-the-art review and annotated bibliography of system analysis techniques applied to reservoir operation. Texas Water Resources Institute, Texas A & M University.

Yeh, W., Becker, L., 1982. Multi-objective analysis of multi-reservoir operations. Water resources research 18 (5), 1326–1336.